Accident Analysis of Software Architecture in High -Reliability Systems: Space Based Infrared System Software Problems

Prepared By

Sujatha Mohanram

Accident Analysis of Software Architecture in High -Reliability Systems:

Space Based Infrared System Software Problems

Sujatha Mohanram

Submitted in fulfillment of the requirements for the

Degree of MSc Research

School of computer science

University of Glasgow

June 2017

The views expressed in this thesis are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

Abstract

The accident analysis of SBIRS program is conducted by gathering information for 15 years to understand the cause of the accident. The program had series of failures, workarounds were developed incrementally to solve the incidental problems over the years. This resulted in major failure in thermal vacuum testing. The architecture was reassessed, the new architecture so adopted was the wrong architecture. This is the accident this research has analyzed. The cause of the accident is analyzed thoroughly to understand the circumstances in which such an architecture was adopted.

A System analysis of the environment was conducted to understand the accident circumstances and an accident analysis was conducted to understand the influence of the systemic failures of the wrong architectural decision which is the accident analyzed. A comparative study of accident analysis methodologies was undertaken to derive the best-suited method for accident analysis. A systemic accident analysis method STAMP, which analyses the accidents caused by the influence of the environment was considered as the best fit.

The STAMP accident analysis method was adopted to understand the accident in detail. The accident analysis was performed based on the reports gathered from GAO, DOD and other sources and was confirmed for its completeness and accuracy from GAO. STPA process was adopted to conduct accident analysis in three stages – identifying control structures, changes in control structures and dynamic process model. STAMP accident analysis was improved by adding context as an additional factor.

Accidents with context as the cause of the accident were analyzed to understand the possible solutions. The realization of the importance of context as accident cause was understood and the need to enhance the accident analysis model was realized. By adding context as part of the process that needs to be transferred to ensure successful completion was suggested. An organizational model that has been successful in assessing the accidents due to the context in the different domain was studied and was suggested to be adopted as preventive accident analysis model. Finally, the wrong architectural decision being the accident is contested and argued as the accident, as currently such decisions are not considered as an accident in the industry.

This research has identified the cause of the accident to be the context in which organizations were operating. The solution suggested is to stabilize the context in one organization and replicate the stabilized context around the organizations involved in the program. The solution identifies contextual enhancement techniques used in health and safety management to build a positive culture in the organization.

Thus this research has contributed towards analyzing the architectural failure in SBIRS program by identifying an accident analysis method that best suits the case study, applied the accident analysis to the case study to understand the cause of the accident. A recommendation of enhancing the factors in accident analysis was suggested and an accident prevention technique was recommended and a process to adopt this technique was suggested.

This research has led to two further recommendations for future work. An architectural technique which would create the framework of components to prevent future architectural accidents such as this case study will be followed up. And a process to successfully pass the context in order to prevent accidents caused by organizational context will be taken further.

This research is structured to understand the problem, analyze the problem using specific accident analysis methodology related to the domain detailing the accident, comparing different domains with the similar accident cause and finally recommending an accident prevention technique which had been successful in organizations.

Contents

Abstract 2

List of Figures..........5

Acknowledgment..........7

Author’s Declaration..........8

Abbreviations..........9

Thesis Structure – Thematic outline..........10

Introduction..........14

Chapter I Comparative study of accident methodologies..........19

1.1 Traditional Accident Analysis Methods..........19

1.2 Modern Approaches to Accident Modelling..........23

1.3 Rasmussen’s Sociotechnical Framework for Risk Management..........26

1.4 System Theoretic Accident Model And Process..........29

1.5 Formal Methods for Accident Analysis..........31

Chapter II SBIRS program Analysis..........34

2.1 System analysis..........34

2.2 Accident Analysis..........39

Chapter III Systems Theoretic Accident Model and Processes..........43

3.1 STAMP..........44

3.2 STAMP based Process Analysis (STPA)..........47

3.3 SBIRS Architectural Accident..........50

3.4 STAMP analysis of SBIRS Architectural Accident..........59

Chapter IV Recommendation of accident prevention..........115

Chapter V Architecture Analysis..........119

Conclusion..........122

Recommendations for Future Work..........125

Appendix..........126

Reference..........127

List of Figures

Figure 1 Sequential event-based model [Qureshi, 2008]............20

Figure 2 Time ordered chain of events model [Qureshi, 2008]............21

Figure 3 Fault Tree Analysis[Qureshi, 2008]............21

Figure 4 Failure Modes Effects Analysis [Toolbook et al., 2017]............22

Figure 5 Charles Perrow's Seminal Model[Qureshi, 2008]............23

Figure 6 Reason's organizational model of system accidents[Qureshi, 2008]............24

Figure 7 Reason's model and event chain model[Qureshi, 2008]............25

Figure 8 Structural Hierarchy and System Dynamics[Qureshi, 2008]............26

Figure 9 AcciMap Accident Analysis Technique[Qureshi, 2008]............27

Figure 10 Kings Cross Fire Accident............28

Figure 11 King's Cross fire accident – cross causal factor relations............28

Figure 12 STAMP component interaction context............29

Figure 13 STAMP analysis of Walkerton water contamination accident[Leveson et al., 2003]............30

Figure 14 STAMP Model [Leveson, 2011]............45

Figure 15 STAMP core principles............46

Figure 16 Classification of control flaws leading to accidents [Leveson, 2011]............48

Figure 17 Flight Software Architecture[GAO, 2008]............50

Figure 18 Recommendations of flight software architecture[GAO, 2008]............51

Figure 19 STAMP SBIRS Control Structures............62

Figure 20 Represents the control structured that had disappeared over time and become ineffective. (Dotted lines)............77

Figure 21 Risk-Dynamic process model............88

Figure 22 Requirement – Dynamic Process model............89

Figure 23 Design – Dynamic Process Model............90

Figure 24 Development – Dynamic Process Model............91

Figure 25 Test – Dynamic Process model............92

Figure 26 Program – Dynamic Process Model............93

Figure 27 Accident – Dynamic Process Model............96

Figure 28 Dynamic model of SBIRS Architectural failure............98

Figure 29 Organisational Context............99

Figure 30 Recommended organizational context enhancement............105

Figure 31 Positive safety culture[Institution of Occupational Health and Safety, 2015]............116

Figure 32 Maturity Model for safety culture[Institution of Occupational Health and Safety, 2015]............117

Figure 33 Total Quality Management for change process[Institution of Occupational Health and Safety, 2015]............118

List of Tables

Table 1 Interaction failures mapped to controllers 86

Table 2: Finding of this research 114

List of Reports

Report 1: GAO Assistant Director, familiarity with the SBIRS program........ 126

Report 2 : Assistant Director, GAO- Confirmation statement on SBIRS project........ 126

Acknowledgment

To my life-coach, my farther Wing. Cdr. R.T. Ramanujam (retd): because I owe it all to you for all the support and confidence in me. For all the stupid questions I might have asked, for all the encouraging enthusiasm that you contributed. For being there when I needed the most especially when I thought of bringing knowledge from other streams into my stream, even when others discouraged the idea. All the guidance you imparted to be cautious and persevere to reach the goal, for all the moral and emotional support you gave Many Thanks for being you!

My eternal cheerleader, my brother: I miss our interesting and long-lasting chats on this research. My forever interested, encouraging and always enthusiastic brother, always keen to know what I was doing and how I was proceeding. I loved all the momentous moments spent during the course of my study, all your screams of joy whenever a significant moment was reached and also just your general impudence which encouraged me to think deeper in research. And of course thanks for all the knowledge and guidance you gave during the course of my study, which I am ever grateful!

And finally, last but by no means, the least, also to Professor Christopher Johnson for guiding me through this course since last three years.

Thanks for all your encouragement!

Author's Declaration

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution and to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Glasgow and where applicable, any partner institution responsible for the joint award of this degree.

I give consent to this copy of my thesis when deposited in the University Library, being made available for loan and photocopying, subject to the provisions of the Copyright Act 1968.

I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission had been granted by the University to restrict access for a period of time.

Abbreviations

AFSPC – Airforce Space Command

AFOTEC – Airforce Operational Test and Evaluation

AFROC - Air Force Requirements Oversight Council

AFI-99-101 - “Developmental Test and Evaluation” Airforce Instruction 99-101 (November 1, 1996)

AFM99-113 - Air Force Manual 99-113, “Space Systems Test and Evaluation Process Direction And Methodology for Space System Testing,” May 1, 1996

AI&T – Assembly Integration & Testing

CRD - Capstone Requirement Document

COEA - Cost and Operational Effectiveness Analysis

CONOPS - Concept of Operations

CAIG – Cost Analysis Improvement Group

CAIV - Cost as an Independent Variable

CRIMS - Cost Risk Identification and Management System

CGRA - Common Gyro Reference Assembly

DAB - Defense Acquisition Board

DAE - Defense Acquisition Executive

DOT&E – Director Operational Test and Evaluation

DCM -Design Compliance Matrix

DSP - Defense Support Program

DCMA - Defense Contract Management Agency

Directive 5000.1 - Defense Acquisition, Feb 23, 1991

DOD – Department of Defense

DOD Instruction 5000.2 - Defense Acquisition Management Policies and Procedures Feb 23, 1991

EMD –Engineering and Manufacturing Development

Effectivities - milestones at which an incremental system capability is delivered by the developer and accepted by the operator

EAC – Estimate at completion

EMI – Electro Magnetic Inference

FRB - Failure Review Board

GAO – Government Accountability Office

GEO – Geosynchronous satellite

HWIL – Hardware In the Loop

IHC – Interim High Elliptical Orbit Capability

IATO - Interim Authority To Operate

IOT&E –Initial Operational Test and Evaluation

IBR - Integrated baseline reviews

IPT - Integrated product team

IPR - Interim Progress Reviews

ICE - Independent Cost Estimate

IOC – Initial Operational Capability

J5 – Joint Staff -Strategic Plans and Policies

JROC - Joint Requirements Oversight Council

KPP - key performance parameters

MNS - Mission Need Statement

MR – Management Reserve

OSD - Office of the Secretary of Defense

ORD – Operational Requirements Document

RRW - Relative Risk Weighting

RVL - Requirements Verification Ledger

RTMT - Requirements Traceability and Management tool

RRG - Requirement Review Group

RSE - Requirements Systems Engineering

SAR – System Acceptance Review

SBC - Single Board Computers

SBIRS – Space Based Infrared Satellite

SPO - SBIRS Program Office

STAR - System Threat Analysis Report

SAMP - Single Acquisition Management Plan

SMC -Space and Missile Systems Centre

SRA - System requirements analysis

SWF - Senior Warfighters Forum

SEIT- System Engineering Integration Team

STAMP – System Theoretic Accident Model and Process

STPA – STAMP based Process Analysis

TEMP - Test and Evaluation Master Plan

TRL- Technology Readiness Level

USAF -United States Air Force

USSPACECOM - United States Space Command

Thesis Structure – Thematic outline

Change management is a continuous assessment process. Most organizations face the problem of adopting and adapting to the change. The change in the SBIRS case study is to adopt the new architecture of single core processor. This research is to argue that new architecture so adopted is the failure of the change management process.

Software architecture accident analysis. Architecture is the initial phase of the development cycle, where the system of identifying the solution is performed to the best efficiency by drawing a plan. This case study has failed to draw a successful plan for the development of flight software which has resulted in major failure whereby the entire process of architectural planning had to be repeated. The new architecture was still a failure plan.

Accident analysis is undertaken when the end result is successful. The end result of the program governs the initiation of accident analysis. The successful completion is considered as an overall success of the program. In this case study the SBIRS program had success outcome, so the program was not assessed for the failures that this program had gone through. This case study was chosen to understand the failures to learn from them even when the end result of the program is successful.

Accident analysis is not undertaken in the industry. In general, accident analysis is considered only when there is a loss of life or a huge loss of resources. Software industry like government services, which is reliant on software services for its operational success, lacks knowledge to understand the complexities that the software imparts to their main stream failures. These failures are normally not dug deep, efforts are to blame the software process. There are many software process improvements over the years but the architecture of software is never blamed. There is no prominence given to accidents that are due to architecture as it is the decisions taken at the very start. The accident analysis is to identify the failures and to learn from them. In this case study, the loss is time, cost and software architecture. The software architecture is the cause of failures that had followed with defects and redesign efforts.

Industry gap. There needs to be a strict architectural framework for the software applications to formulate a well interacting successful software to be delivered on time and in the budget. This case study has conducted accident analysis to learn from the failures of architecture which has led to huge loss of money and loss of time. In order to solve this problem, an architectural framework which holds the components with their pre-defined interaction and policies that govern entry and exit criteria to this framework is suggested.

Accident Prevention techniques. "Blue Print" of software architectural framework which holds the components with the interactions assembled and their foundations laid for the choice of technologies to bounce against a strong framework with is an attempt to prevent accidents due to architectural failures. STAMP accident identification factors were enhanced to include context as a factor to be analyzed. A preventive technique of successful context is distributed among the organizations to prevent accidents such as this case study.

Identification of causes of the accident. The case study has undertaken accident analysis to understand the causes of the failures leading to the accident. The cause of the accident in the case study was analyzed as the 'context' of the organizations involved. The failures were due to initiatives brought in the SBIRS program and this change process was not continuously monitored which resulted in the architectural failures and subsequently resulted in adopting an architecture with single core processor which would lead to failures in future.

Analysis of accident (process). The case study has gathered information for 15 years of this program to understand the accident circumstances. The analysis of the program was conducted with GAO's participation in assisting the research by validating the information gathered for analyzing the cause of the accident. NASA had guided the research by providing detailed architectural evolution history and validating the recommendations put forward in this research.

Literature review. The information was gathered from GAO, DOD, SBIRS program office, Under secretary's office for Defense, Lockheed Martin, Airforce, NASA GSFC, NASA JPL and published news articles. NASA had provided the latest architectural documents and historical perspective of failures discussed in the Software Architecture Review Board (SARB).

Methodology – STAMP. This case study was reviewed with the STAMP methodology to understand the failures that led to the accident of adopting a wrong architecture of single core processor.

The environment of SBIRS program. This case study has analyzed the SBIRS program with a vision of identifying the influencing factors of the development lifecycle of the flight software. The organizations involved were facing problems due to the pressure of stakeholders, political influences, military constraints, contractors, coordination problems, relaxed procedures and many new changes brought forward to influence the outcome which led to failures to cope with the changes were studied.

Architectural failures. This case study has analyzed many architectural failures in the program to understand the circumstances in which the architectural decision of adopting single core architecture for flight software was recommended. This is the accident this research has analyzed.

SBIRS program. This case study has brought forward interaction failures between the organizations involved, coordination failures, change initiatives, requirements problems, architectural problems, design problems, test problems, assurance problems, budget problems, planning of project problems, military standards causing problems, lapse in adhering to the military standards, process failures, waivers, environment to encourage lethargy, short-term cover-ups – workarounds, cost escalations and delayed schedules. The encouraging factor to choose this case study was the architectural failure which was astonishing to see the SBIRS program adopt single core processor for flight software architecture which is the primitive stage of the architectural design.

This research is structured to show ***the effect of changes in the continuous assessment process*** in the SBIRS program analysis section by explaining the problems in the environment and ***the failures in software architecture*** as the architecture that is considered as an accident is shown in Accident analysis section explaining the details of the accident that is examined in Chapter II. ***The accidents that occur when the resultant is a success*** are not studied, so this case study has studied the accidents that had resulted in success. In order to understand such accidents, a methodology is chosen. The methodologies pertinent to this accident analysis is analyzed for relevance to this case study in Chapter I.

The accident is not considered as an accident, in this case study the architectural failure is considered as an accident. Accident analysis of the case study is conducted for the architectural accident using STAMP accident analysis where the cause of the accident is derived from the systematic analysis of the accident. It is argued as an accident in chapter V.

The ***industry has a gap*** in the prevention technique for accidents that are caused by the context. So a ***prevention technique*** is to be formulated. An enhancement to the STAMP model is suggested for analyzing the accidents with the context as one of the factors to be considered while analyzing accidents in Chapter III.

The ***cause of the accident was analyzed*** for the influence of the change process over the system. A recommendation for accident prevention for the failures due to contextual factors are analyzed in Chapter IV. ***The accident was highlighted*** to GAO. A detailed overview of why such an accident is considered is explained in Chapter V. The ***architectural accident*** is the sole cause of the failures in this program. This research has led to future research recommendations which are detailed. An analysis of GAO's recommendations to this research is appended in Appendix.

Introduction

The software industry has seen many failures in the software development in terms of meeting the requirements, cost overruns and schedule delays. Software development has been unpredictable in terms of development, performance, integration, and utilization of software for the intended need. The assured quality of the software is always in question, the quality of the software is unpredictable. Failures in software development need to be analyzed for improving the quality, predictability, and delivery on time.

The software industry has to learn from failures to improve on meeting the expectations of the stakeholders. To achieve this, accident analysis has to be performed to understand the failures of software to match the expectations. Accident analysis is considered as a process to be conducted after the accident has occurred. There is no accident analysis undertaken in software industry unless the result of the project is a failure. As a result, most of the accidents go unnoticed and there is no knowledge gained as the resultant of the project dominates the initiation of accident analysis. In this case study accident analysis is performed even when the end result of the program was successful and the accident was not noticed.

In general software architecture is not blamed for wrong decisions adopted. The blame is borne by other teams such as development and test teams. In this case study, it is been argued that the choice of single-core architecture was not the right decision. An accident analysis is conducted to understand the importance of architectural decisions in the success of the project. The decision of adopting single core architecture was to see the failure that will definitely occur to consume more resources, for the time to shed light on this failure which might have caused enormous loss of time, money and quality by evolving through the fault initiated at the architectural phase.

STAMP accident analysis is adopted in the case study to show the success of the preventive accident analysis. As the change is a continuous process, organizations should assess the organizational context continually to avoid loss of resources. A process to assess the organizational context is suggested which is based on the proven technique used by Health and Safety organization. This case study was adopted to prove that the failures have to be analyzed to learn from, even if the final outcome is successful.

SBIRS program had been restructured several times due to schedule delays and cost increases resulting in revised program goals in 2002, 2004 and 2005.[GAO, 2008] There were many challenges in the program due to technologies and software. This research is to understand the successes and flaws in the system to navigate through the deep sea successfully as this program had managed to fly high in 2011[Mccaney, 2014]by successfully launching the first GEO 1 satellite with higher performance efficiency[LockheedMartin, 2012].

The concerns were due to architectural failures leading to defects which resulted in a tremendous amount of pressure on the development team and operations team to hold the bag for the failed architecture. In 2007 GEO satellite underwent thermal vacuum testing and had major failures due to architectural issues[GAO, 2008]. This failure was deducted after 12 years of program inception[GAO, 2008]. There were many interim architectural issues found in different stages of the program, there were workarounds[GAO, 2003] developed to have temporary relief from the situation which had mounted up to this stage of thermal vacuum test failure in 2007[GAO, 2008]. At this stage of the program, there had not been any way out of the situation as there were defects mounting from every component planned to be developed in flight software. The only way out was to redesign the software architecture with all the defects encountered resolved in the new design.

The redesign effort undertaken was after a thorough analysis of the architectural problems, to climb out of the issues already presented due to architectural problems and to build a safe architecture, the architecture was to place all the software components in one processor[GAO, 2008]. This approach is questioned in this research as it was a wrong decision to go back to the primitive stage of architectural design. The earlier version of the architecture was to have flight software deployed in two processors having the advantage of the distributed system.

Although this architecture with the new design could have resulted in another catastrophe, fortunately, there seems to be an explosion in the evolution of architecture after this stage which resulted in using advanced architectural principles in flight software. The flight software in the current state is in a distributed system with Mission Data Systems architectural[Feller, Gluch, and Woodham, 2010] approach along with core Flight Software systems[Mccomas, 2012] approach which is widely appreciated for its achievements in architectural success beyond measure of any successes this project might have had in the history of space systems.

This research is structured initially to identify an accident which did not result in final failure. The failure had taken the toll by wasting resources, time, and quality of the software. The identification of accident is understood in the context of other organizational failures in Section 2.1. The accident context is explained in the Section 2.2. The architectural failures are not considered as accidents, so this research has identified this architectural failure by surveying various other methodologies of identifying the cause of the accident in the Chapter I Comparative study of accident methodologies. As the accident is architectural, this accident has the preset context of failures in the program, this accident needs to be analyzed with a methodology which comprehends the architectural context along with the wider program context. A systemic method was adopted to analyze this accident using STAMP accident model in the Section 3.1.

As STAMP, is a hierarchical model, this accident is considered as spread over different areas of concerns, were the hierarchical model is the right model to be chosen with contextual factors being analyzed in a systematic manner using STPA process in the Section 3.2. The architectural accident is detailed for the accident analysis in the Section 3.3. The hierarchical levels of control are identified which were responsible for the architectural accident detailed in the section 3.4.1. The change is a continuous process which was not adapted to by the program and the failures in adaptation was the resultant of the architectural accident which is detailed in the section 3.4.2 Changes in Control structure leading up to the Accident. As all organizations are in a dynamic equilibrium in response to the change, the dynamic context would depict the actual cause of the accident which is detailed in the section 3.4.3.

The findings of dynamic context are realized as an unusual cause of the accident, so other accidents which had similar cause were analyzed in the section 3.4.4 “Context” As Cause of Accident in other case studies. It was understood that other accidents have also not initiated any prevention techniques, so an accident prevention technique is suggested in the section 3.4.5. It is realized that this accident cause is not been learned from, which leads to a worrisome outlook to be continued even with this accident, so a prevention process is suggested to adopt the accident prevention technique recommended in the Chapter IV.

The architecture is considered to be the accident as it would lead to further loss of resources and time, this is argued in the Chapter V. This research has resulted in an invention of a prevention technique as “Blue Print” of architecture and a process to adopt the change as detailed in the section Recommendations for Future Work. The detailed process of accident analysis is explained further.

STAMP analysis takes step by step analysis of the accident. The first stage of analysis is to understand the hierarchy of control structures involved in the accident. The next stage of the accident analysis is to understand the flaws in the process and enforcement of constraints. The motivation to change and to adopt by the organization to the control constraints is also analyzed here. The next stage is to map every deviation in the whole system, analyze the effects of one system over other, how each system had contributed towards creating an expectation of conformity to the overall norm. This norm is identified as the cause of the accident. In STAMP analysis, a detailed accident analysis is undertaken while considering the whole context of the accident.

STAMP analysis had brought forward an intrinsic factor of the change process by decoupling every aspect of the accident in a systematic process of revelation as we walk through it. This revelation had been a shock at every stage in the process of accident analysis. As it reveals its true self, has an enormous world opening up in front of the STAMP process which encourages accident analysts to understand intricate aspects of the STAMP analysis. This revelation does bring forward one’s self-image to the level of self-actualization which promotes utilization of every resource that had been gathered over the run which increases one’s knowledge about the accident. STAMP analysis is one which has erupted the cause of the accident to this state.

This research has contributed to identifying the accident among a huge pile of information on different organizations involved. This accident had gone unnoticed and was never analyzed, as failure was not the end result. The accident was analyzed using STAMP analysis, after understanding the different techniques used in accident analysis. The cause of the accident was found to be the context of the organization in which it was operating.

This research has analyzed various failures in the program, along with schedule delays and cost overruns. This research has chosen to analyze the accident at the point of the thermal vacuum test failure in 2007. After this major failure, the architecture was reassessed in the light of all the failures thus far, and in an attempt to solve all the problems by simplifying the architecture, the architecture with single core processor and applications deployed were not on distributed architecture was adopted. This has arisen the accident of not adopting multicore architecture and distributed application architecture. The accident here is the adoption of wrong architecture.

A prevention technique is recommended which had been successfully implemented in other domains. This prevention technique is thoroughly analyzed to understand the factors involved in the accident are prevented. The process of adopting this prevention technique is also detailed to continuously improve the context. This adoption of prevention technique is new to this domain and has been recommended from the theoretical perspective.

This thesis is structured to reveal the accident of SBIRS program analysis in the second chapter, a comparative study of accident analysis methodologies is undertaken to understand the various approaches to the accident analysis in the first chapter, STAMP analysis is undertaken for SBIRS project in the third chapter, the recommendations from the accident analysis is suggested in chapter four and the Architectural failure is explained in the fifth chapter.

In the next chapter, an analysis of methodology to be adapted to perform the accident analysis is discussed. This methodology has to highlight the flaws in the system and bring the actual facts of the accident to light. These methodologies have specific strengths in the analysis of the accident, so a thorough analysis of the accident analysis methodologies is undertaken to understand best-suited methodology to this accident scenario.

Chapter I Comparative study of accident methodologies

Rasmussen in the context of risk management has asked whether '*we actually have adequate models of accident causation in the present dynamic society?*' He argues for a '*model of behavior shaping mechanisms in terms of work system constraints, boundaries of acceptable performance and subjective criteria guiding adaptation to change*' [Feller, Gluch, and Woodham, 2010]. The adequacy and suitability of accident investigation models continue to be open for academic deliberation.

Accident analysis could be analyzed under traditional accident analysis and modern accident analysis. The accident analysis methodologies will be analyzed to understand the most appropriate model to implement in the case study. The case study is based on architectural accident being developed in a complex environment. As the complexity is projected at the very on-sight of the accident, the complete analysis of the history was required to understand the causes of the accident. The accident had unbelievable scope to the architectural analysis, which generated the interest to delve deeper to understand the very scope of the accident. This accident analysis is done for the period of 15 years, understanding what happened during this period to lead to such an accident creating a wider context for the accident.

1.1 Traditional Accident Analysis Methods

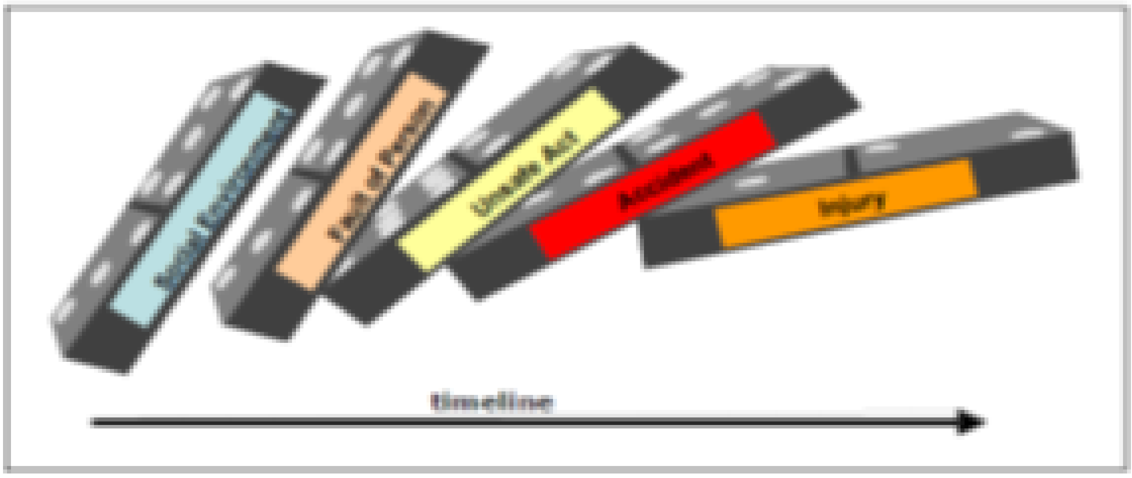
An example of King's Cross underground station fire accident will be considered to analyze various methods to understand the applicability of SBIRS case study.

King's Cross Underground station fire accident

"While several minor escalator fires had occurred previously and had been investigated, apparently no one in the organization seriously considered the fact that a major escalator fire was a possibility – consequently, as the inquiry states, little effective action had been taken on the warnings provided by the minor fires. Similarly, the inquiry also reported that there were serious flaws in the managerial and organizational responsibilities and accountability for safety with virtually all aspects of the organization thinking passenger safety was someone else's responsibility." - Department of Transport (1988), Investigation into the Kings Cross Underground Fire, London: HMSO

1.1.1 Sequential Event based Models

Sequential event-based model is based on the sequence of events to deduce the causal factor from the five sequences, social environment (those conditions allow to take risks); the fault of the person; unsafe acts or conditions (poor planning, unsafe environment, hazardous environment); accident; and injury (*Figure 1)*. The model depicts only single cause for an accident whereas accidents normally have multiple factors[Qureshi, 2008].



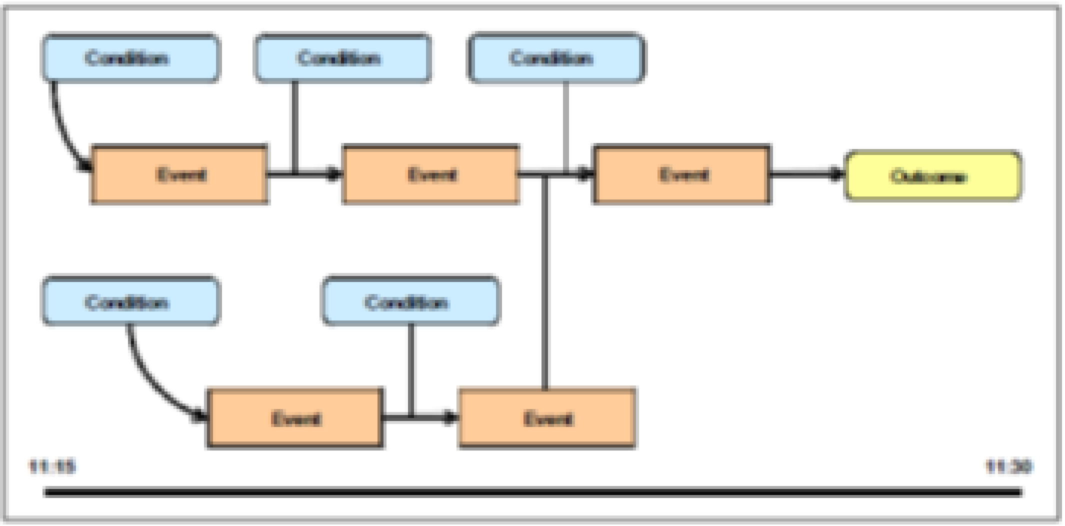
Heinrich's Domino Model of Accident Causation

Figure 1 Sequential event-based model [Qureshi, 2008]

In King's cross accident, domino theory which advocates single cause for an accident is proven inadequate. There were multiple causes of failures to this accident such as management's inadequate action, inadequate preventive measures undertaken after the minor fire accidents, etc.

1.1.2 Time-ordered chain of events model

Events based on time are recorded as a linear model, they are ordered based on time and the event before the accident is blamed as the cause of the accident (*Figure 2)*. This is not appropriate as there could be many reasons for the accident[Qureshi, 2008].



Activity events and outcomes for two actors including events (Ferry, 1988)

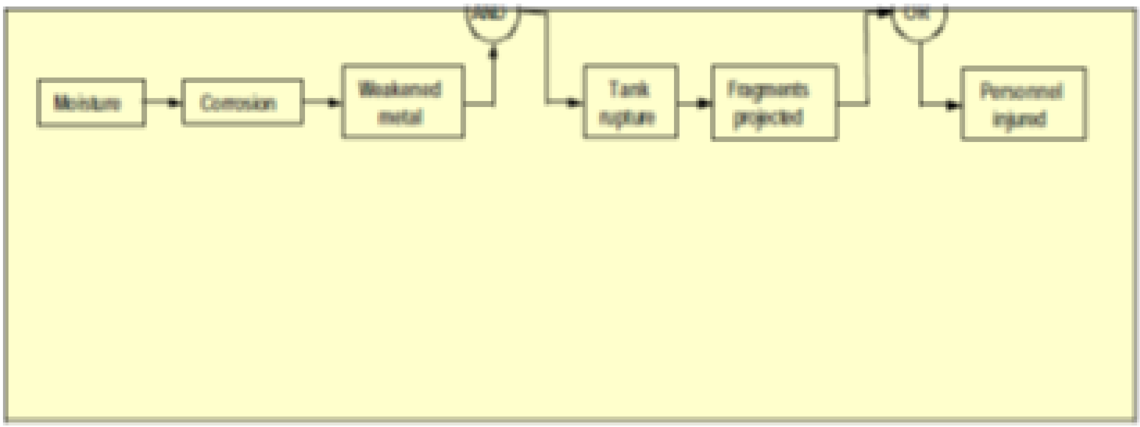
Figure 2 Time ordered chain of events model [Qureshi, 2008]

In King's cross accident, there were multiple sequences of events which related to the cause of the accident. Such as unsafe act of a person in the lift, managerial incompetency to safeguard the welfare of the passengers, etc. This proves that the previous stage before the accident cannot be the only cause of the accident.

1.1.3 Risk Analysis Model

1.1.3.1 Fault Tree Analysis

Fault tree analysis is a predictive hazard analysis tool to determine the hazardous situation at a given point in time. Fault tree model is a snapshot of the state of the system at a given time (*Figure 3*), but it does not convey any notion of time delay or time ordering[Qureshi, 2008].



Fault-Error-Failure Model (Leveson, 1995)

Figure 3 Fault Tree Analysis[Qureshi, 2008]

In King’s cross accident, a major fire in the lift is a top-level event and the lowest order event would be the actual cause of the fire (cigarette bud). As this model does not convey the notion of time delays like time to fetch the fire extinguisher or time delay in spreading of the fire, decisions taken at the wrong time would lead to unpredictable results.

1.1.3.2 Failure Modes Effects Analysis

Failure modes effects analysis uses forward search based on event chain model when initiating events are the cause of the accident (*Figure 4)*. Failure modes effects model does not consider all the events in a collective manner, they are treated as individual failures[Qureshi, 2008].

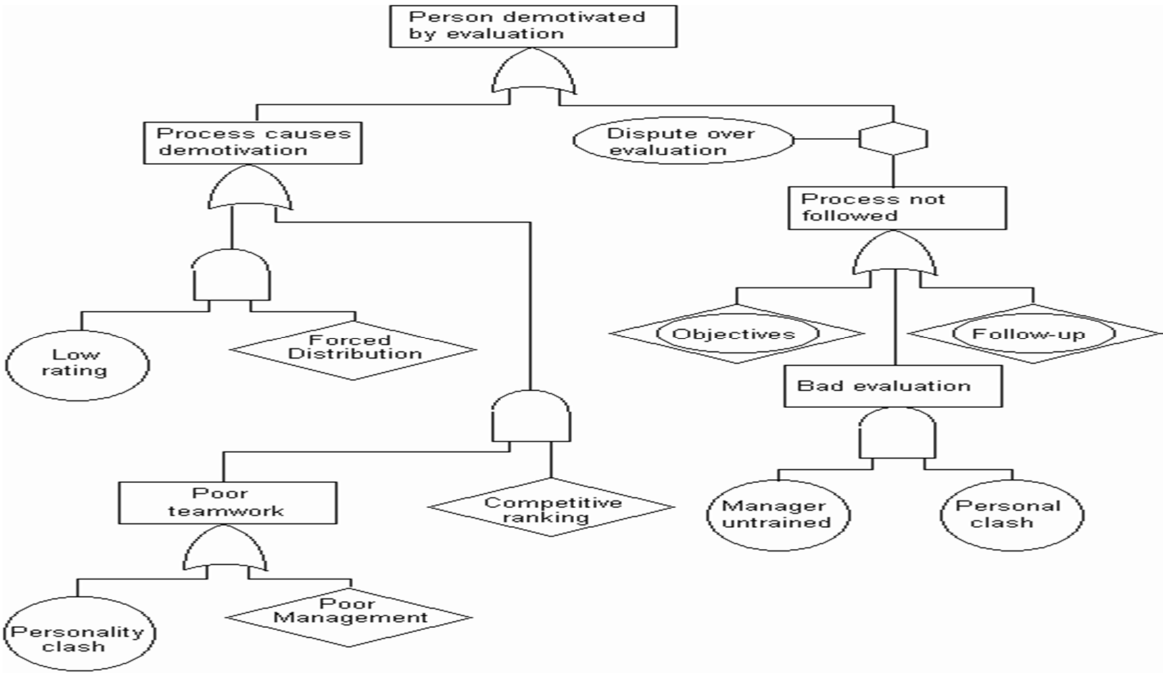


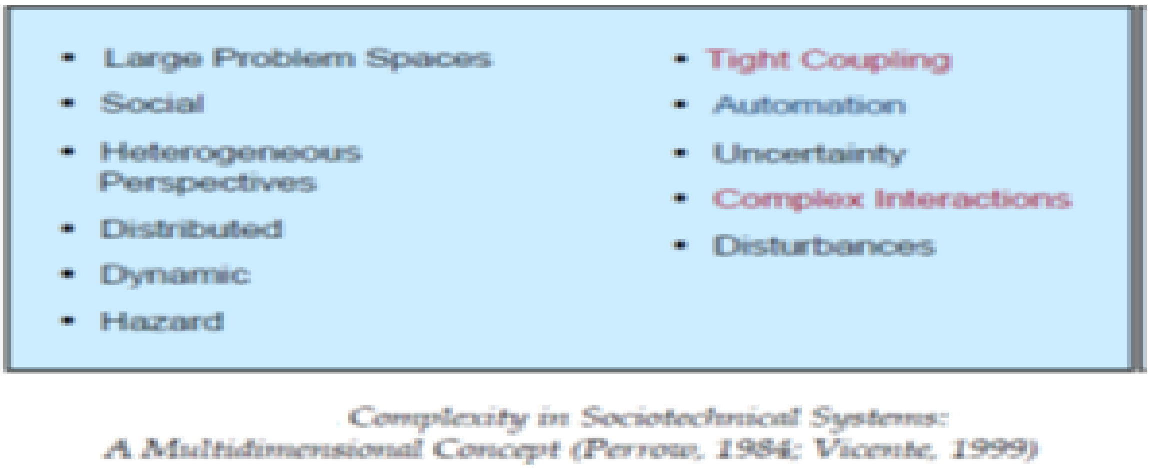
Figure 4 Failure Modes Effects Analysis [Toolbook et al., 2017]

In the King’s cross accident, there were many failures like grease left unnoticed, organizational safety measures were not followed, etc. which would lead to different initiating events (*Figure 4*) leading to failures. As each failure is considered individually, collective failure resulting out of failures are not considered.

1.2 Modern Approaches to Accident Modelling

1.2.1 Charles Perrow's Seminal Model

The main basis of the accident is complex interactions and tight coupling. So the accident analysis analyses the characteristic that makes more prone to the accidents. This accident model considers the multiple accident causes as failures and tightly coupled system that are very dependent on the interacting systems and would have an adverse effect on the interacting components (*Figure 5*). As every system is different and does not comply with the norms of this model[Qureshi, 2008].



- Large Problem Spaces
- Social
- Heterogeneous Perspectives
- Distributed
- Dynamic
- Hazard
- Tight Coupling
- Automation
- Uncertainty
- Complex Interactions
- Disturbances

Complexity in Sociotechnical Systems:
A Multidimensional Concept (Perrow, 1984; Vicente, 1999)

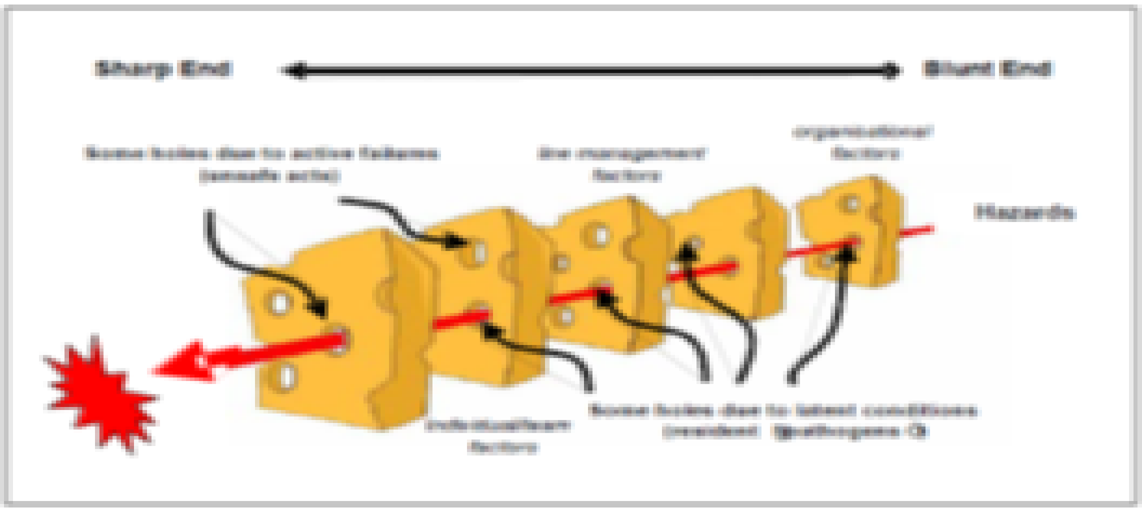
Figure 5 Charles Perrow's Seminal Model[Qureshi, 2008]

In King's cross accident, the tight coupling of the systems like, maintenance of the lift which led to the grease left behind, emergency procedures not properly adopted, etc. are tightly linked to having the adverse effects on the system to cause an accident. The accident methodology could not be used for all systems as every system is different within the King's cross accident.

1.2.2 Reason's Organizational Model of System Accidents

An accident is the interaction of the components where the environment in which it is operating cause the failure when it is destabilized. The cause of the accident is generally that interacts with the system. The dynamics of the system is represented in the form of barriers and safeguards. These barriers support each other as a support mechanism. The Swiss cheese model shows holes in the barriers as accident causes (*Figure 6*).

The accident is depicted as holes in the barriers of cheese when they line up in all layers to cause an accident which does not provide all the causes of the accident and their combinations to result in an accident. This accident model projects only high level of analysis of contributory factors in an accident. Reason's model shows a static view of the organization, whereas the failures are transient i.e. the holes in Swiss cheese are continuously moving[Qureshi, 2008].



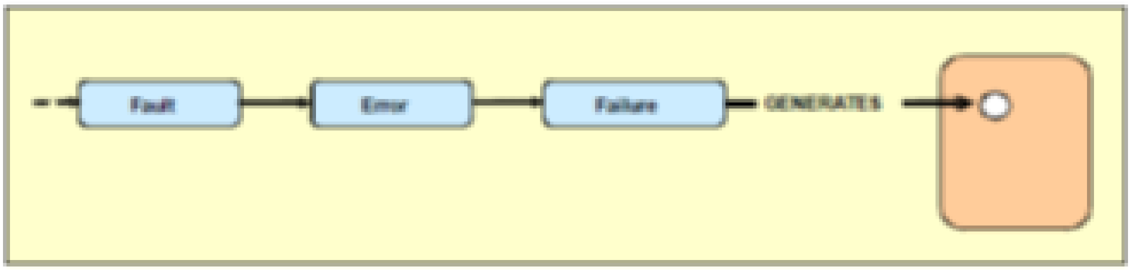
Swiss Cheese Model of Defences (Reason, 1997)

Figure 6 Reason's organizational model of system accidents[Qureshi, 2008]

In King's cross accident, the barriers are the maintenance of lift which had grease, the hole in the barrier would be that it was not cleaned, organizational responsibility would be a barrier and hole in the barrier would be that safety measures were not put in place. These barriers and holes in the barriers may not line up if analyzed after there is a situation change, if the maintenance team would have cleaned the grease but if the nature of cleaning was not specified in the manual, then the holes in the barriers would shift or it might even shift the barrier if the training had not been provided at all by personnel development team. Thus this model does not take into account the dynamic nature of the accident.

1.2.3 Reason's model and event chain model

The accidents in socio-technical systems are a combination of factors meshed into the complex causal network with hierarchical levels in an organization. The technical and organizational issues need to be simultaneously considered. This integration of Reason's and the event chain model provides the lineup of the holes to formulate fault, error, and failure of the accident between system layers (*Figure 7)* [Qureshi, 2008].



An elementary event chain generating a hole in a system layer (Besnard & Baxter, 2003)

Figure 7 Reason's model and event chain model[Qureshi, 2008]

In King's cross accident, there were many event chains that could be created with Swiss cheese model and collated to form fault-error-failure model. But the model is not holistic in approach and linear which fails to incorporate intermediary failures that did not line up.

1.2.4 Systemic Accident Models

The systemic view is considered where the accident is analyzed for the whole system. Thus the accident arises from the degraded interaction between the systems. The system is considered as a dynamic process which continuously changes in response to the environment. Thus accidents are treated as flawed processes and interaction problems in STAMP accident model. Rasmussen considers the contextual factors involved in the organization that creates preconditions for accidents[Qureshi, 2008].

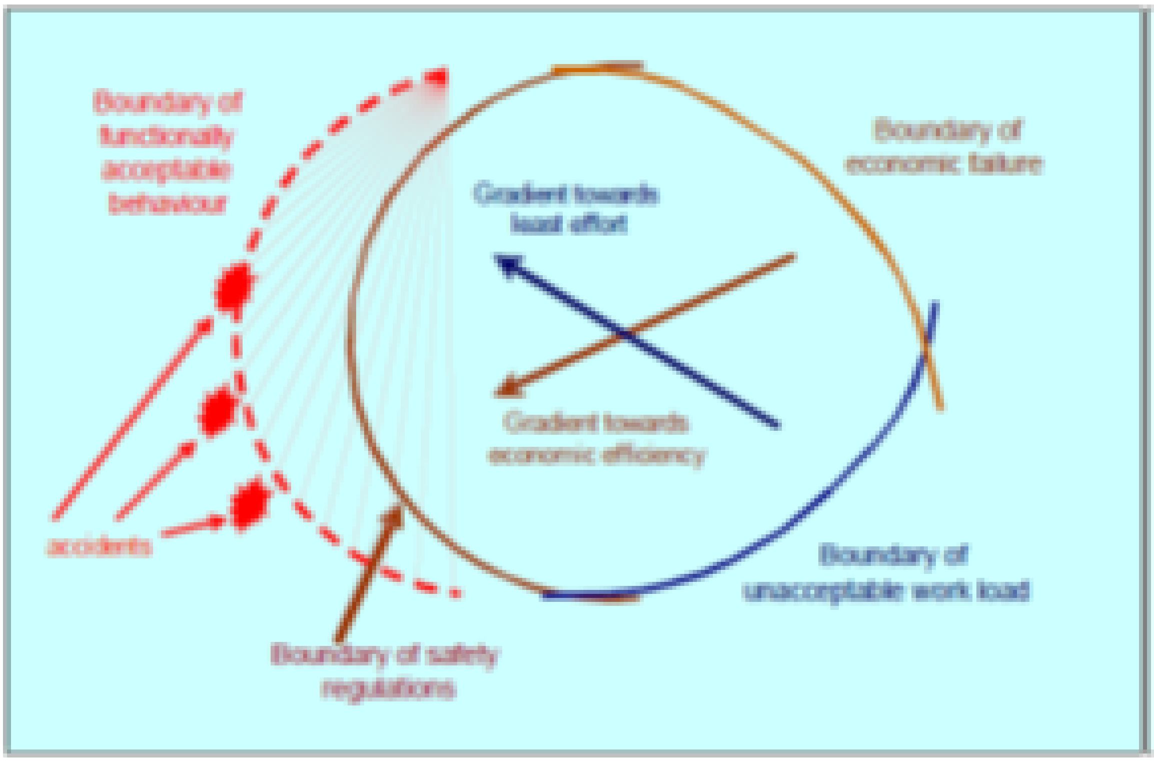
1.2.5 Cognitive systems engineering approach

Cognitive systems engineering takes into account the context in which human and machine interact. It is an understanding of how human and machine function together in the environment than how they interact with each other. Two systemic accident models are developed: Cognitive Reliability and Error Analysis Method; and the Functional Resonance Accident Model. Cognitive Reliability considers human errors in the accident analysis. Functional Resonance model considers the system components that interact with the environmental factors to create an accident[Qureshi, 2008].

1.3 Rasmussen's Sociotechnical Framework for Risk Management

1.3.1 Structural Hierarchy and System Dynamics

This model evaluates the hierarchical structures and their adaptation to the context over time. When this adaptation evolves, the components in the model might get out of the boundaries resulting in an accident (*Figure 8*). This adaptation at every level of the hierarchy would collectively result in a synergetic effect towards the accident. These boundaries for every hierarchy and every actor had to be established and their boundaries have to be secured and guarded for any breach[Qureshi, 2008].


Boundary of functionally acceptable behaviour
Boundary of economic failure
Gradient towards least effort
Gradient towards economic efficiency
accidents
Boundary of unacceptable work load
Boundary of safety regulations

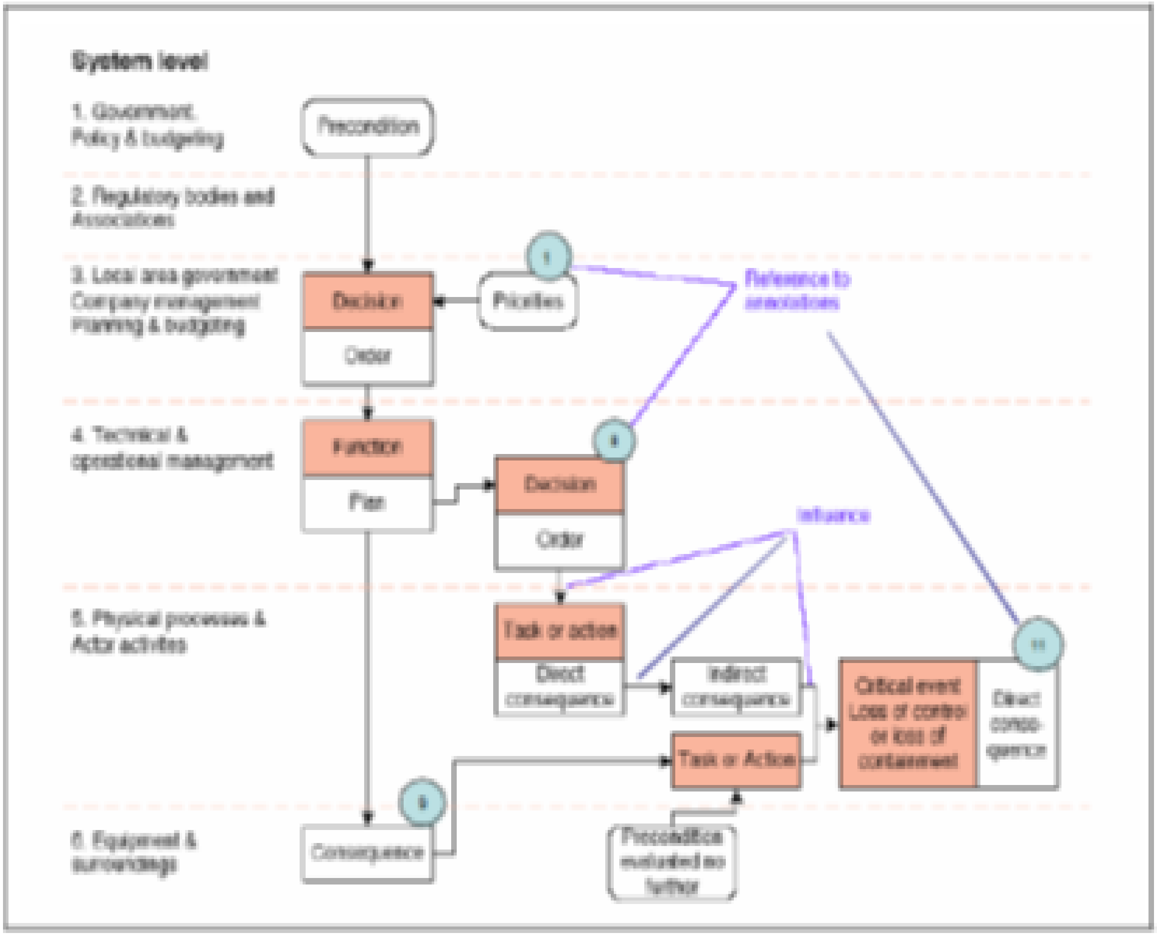
Boundaries of Safe Operation (Rasmussen, 1997)

Figure 8 Structural Hierarchy and System Dynamics[Qureshi, 2008]

In King's cross accident, maintenance team would form the lowest hierarchy, these hierarchies have to be clearly defined. Then identification of boundaries of safe operations would be required. These boundaries have to be made visible to each actor and these should be controllable by the actors.

1.3.2 AcciMap Accident Analysis Technique

AcciMap follows the Hierarchical structure of Rasmussen's framework. It describes the information flow in the entire system (*Figure 9*). This model identifies the interaction flaws between the decision makers and the events leading to accident[Qureshi, 2008]. This model is a linear representation of causal factors which restricts the accident analysis by evaluating the accident in one single area of concern.



AcciMap Structure and Symbols (Rasmussen & Svedung, 2000)

Figure 9 AcciMap Accident Analysis Technique[Qureshi, 2008]

In the King's cross accident, AcciMap each causal chain of events are mapped, then patterns of accidents related to a particular system are analyzed using cause sequence analysis. Set of events are determined by the choice of critical event from cause sequence chart. The critical event that connects the causal tree (potential causes) with the accidents. A vertical analysis of hierarchical levels is conducted which results in an AcciMap as shown in Figure 10.

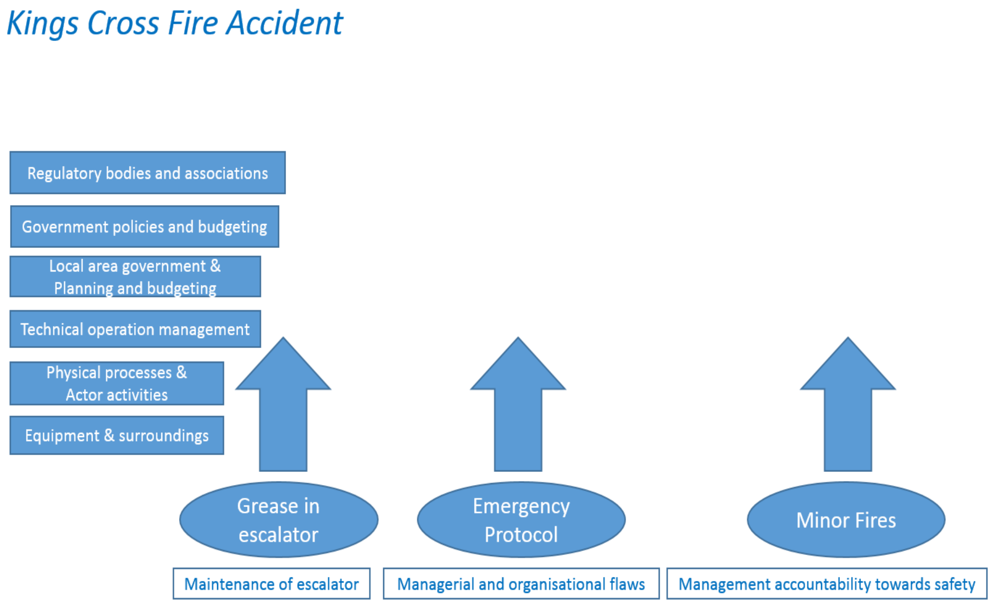

Kings Cross Fire Accident
Regulatory bodies and associations
Government policies and budgeting
Local area government & Planning and budgeting
Technical operation management
Physical processes & Actor activities
Equipment & surroundings
Grease in escalator
Emergency Protocol
Minor Fires
Maintenance of escalator
Managerial and organisational flaws
Management accountability towards safety

Figure 10 Kings Cross Fire Accident

In *Figure 10*, each critical event is represented with sample vertical flow. In reality, there would be multiple AcciMaps created for every critical event and an InfoMap is generated to identify failures in communication between the actors.

AcciMap has a linear approach to the analysis of the accident. AcciMap's linearity causes loss of cross causal factor references in identifying the cause of the accident as shown in *Figure 11*.

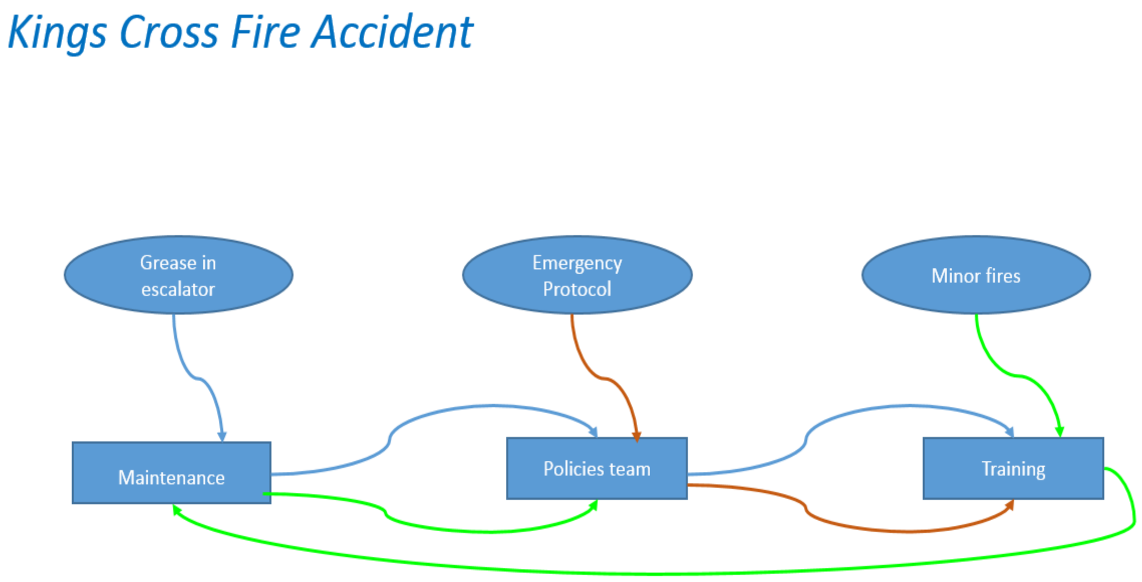


Figure 11 King's Cross fire accident – cross causal factor relations

1.4 System Theoretic Accident Model And Process

STAMP analyses the accident as interaction among the components and the faults that are not adequately handled by the control structures resulting in an accident. This model considers the context of the interaction among the components (*Figure 12*) and when the safety boundaries are violated among the components, accidents occur. STAMP analysis identifies other causal factors such as environmental factors, other actors and their role in accident scenario[Qureshi, 2008]. So STAMP was adopted to the case study to analyze the complex architectural accident in the next section.

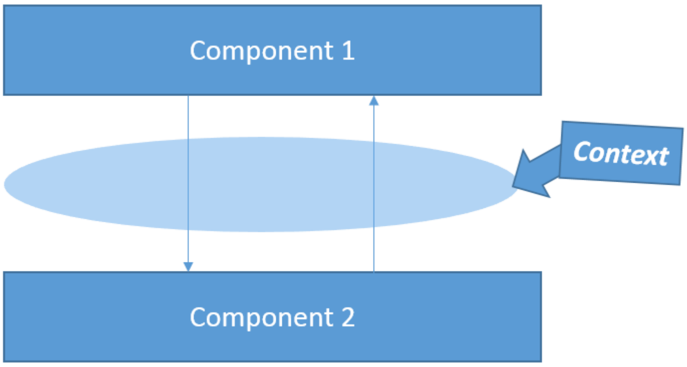


Figure 12 STAMP component interaction context

An example of Walkerton water contamination accident is provided in *Figure 13* STAMP implementation shows control structures (components) and constraint failures (contextual interaction failures).

In King’s cross accident, control structures would be Maintenance department, Underground Management, etc. and constraint failures would be, no safety measures were in place. Thus STAMP analysis clearly illustrates the interaction failures.

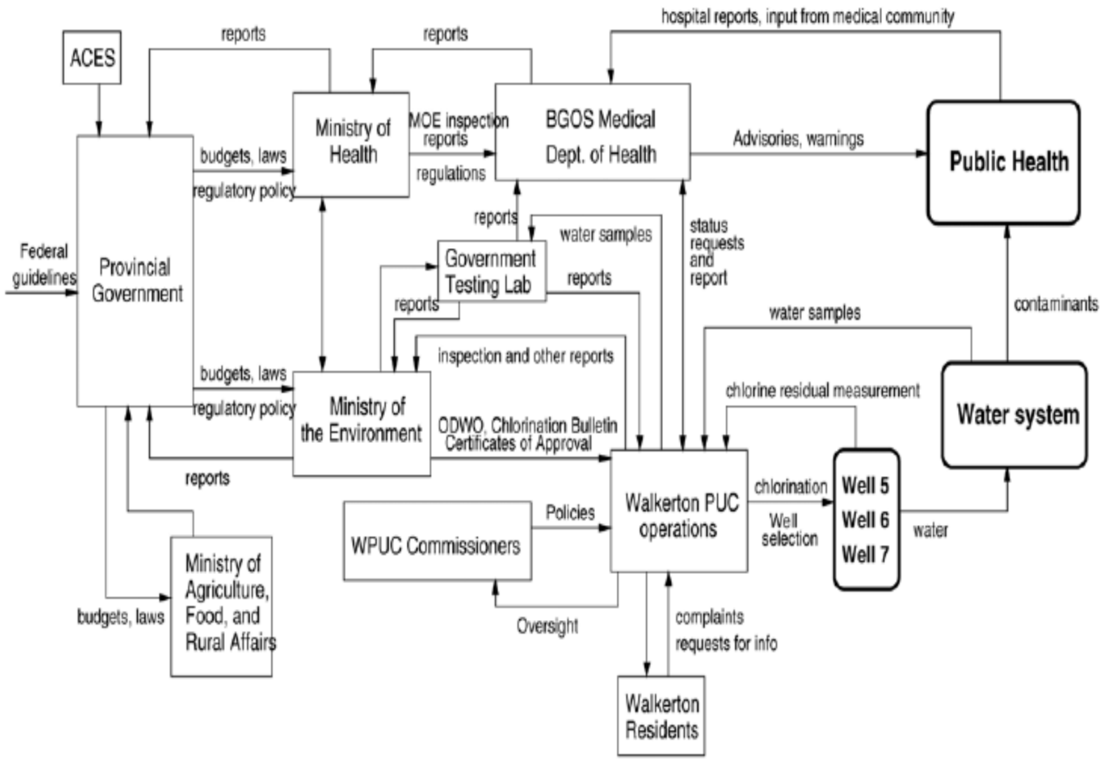


Figure 13 STAMP analysis of Walkerton water contamination accident[Leveson et al., 2003]

The comparative study of Traditional Accident analysis had considered the single cause of the accident and in the current systems, the complexity of interactions are shown to have multiple causes of the accident. In Modern approaches, an evolution of accident analysis has attempted to incorporate multiple causes, while decoupling the system complexity. From Charles Perrow's model to Reason's model, consideration of multiple causal factors had been adopted. Systemic accident models which had incorporated to understand the systemic view of the accident along with the dynamic context.

Rasmussen's sociotechnical model had created hierarchical structures to understand the failures in the system. STAMP had broken the hierarchical taxonomy or classification of the structure into components (control structures) to understand the component failures (constraint) and interaction failures. The STAMP analysis analyses the systems (components) in a systemic manner to reveal many failures as the cause of the accident. The model depicts the evolution of the system over time to degrade and become the failure cause. The model considers missing components which could have contributed to the failure. As STAMP has shown systemic approach to the accident and evolution of accidents over time, this model is preferred to analyze SBIRS accident.

A detailed overview of the STAMP analysis methodology and process is explained in the next chapter. STAMP is implemented in SBIRS case study to understand the missing and failed components and interactions.

1.5 Formal Methods for Accident Analysis

Formal methods are means to describe the accident with accuracy and completeness of the accident to understand all the possible risks that will lead to accidents in order to prevent future failures. The accidents analyzed has a moral responsibility to ensure that information presented should conform to the standardized methodology[Qureshi, 2008]. Burns provides various factors that might influence the accident analysis: *Size* of the accident which might have an adverse effect on the analysis causing ambiguity and omissions; *Structure* of the accident analysis could also lead to confusion; *Validation* will not be possible if a systematic approach is not adopted and the quality of analysis may not be consistent which might lead to sidetrack the analysis from the real problem; *Differing viewpoints* may cause chaos, thus a standardized methodology will help structure the accident analysis in a positive way; *Redundancy* of evidence in a particular area which may not have influenced the accident may gain importance if a strict methodology is not followed; *Imprecision* may lead to repetition of accident and further loss. Accidents should be interpreted in the same way by everybody for that a methodology is adopted to ensure consistency of approach and analysis of the accident; *Concurrency* is important as accidents may occur simultaneously at various places and cumulative effect should be taken into account on top of multiple causes of accident; *Distinguishing Prescriptive* and *Descriptive Behavior* of the accident may cause influence on the analysis by providing prejudice to the analysis thus preventing right prevention action to be taken; *Incompleteness* might lead to misinterpretations of the accident; *Politics* of *Inquiries* may influence the accident analysis in an adverse way, as the real cause could be hidden; *Domain related* accident methodologies which follow the regulations could also have adverse effects when there is no specific proven methodology to follow in accident analysis[Qureshi, 2008].

Accident analysis methodology promotes rigorous reasoning and precision by the methodical construction of formal models which improves accuracy and consistency of accident analysis. An accident analysis should understand the accident and learn from it to prevent future accidents, which encourages identifying all the factors and different perspectives of the flawed processes underlying an accident and ensure that the factual data is interpreted.

Effective accident analysis should focus on social and organizational factors in accidents, system accidents and software errors, human error, and adaptation over time for failure events and reliability engineering techniques to prevent accidents[Leveson, 2004].

A methodology is required to structure the accident analysis. The methodology ensures the accidents to be effectively analyzed. An accident analysis has a moral responsibility to generate right cause for the accident so that the preventive measures could be taken to prevent future failures. A methodology should be proven over the years to adapt the accident analysis based on the methodology in order to attain consistency and accuracy. So the domain related accident analysis methodologies were analyzed for adaptability to this accident scenario.

STAMP analysis is a systemic theory where accidents occur due to external disturbances, component failures or dysfunctional interactions among system components, inadequate control or enforcement of constraints on the development, design, and operation of the system[Leveson, 2004]. In STAMP, the system is a dynamic process with interrelated components that have information flow and control that continually adapts to the changes in itself and its environment[Leveson, 2004]. Thus the accident is described as an adaptive system that fails to meet the complex system goals and values over time due to inadequate control mechanism[Leveson, 2004].

In the King's cross accident, the fire started in the escalator and as a consequence of management failure to prevent this accident 31 people were killed. STAMP analysis of this accident would include Maintenance department, Development and training department, Policies department, Health and Safety organization, etc. that would formulate the hierarchical levels of control structures and the interaction flaws would be that Maintenance department did not advise the employees to maintain the escalator using a set protocol.

Management of Underground did not assess the situation after the minor fires to take preventive action. These failures in interaction would be due to the Maintenance department (controller) that may have issued inadequate or inappropriate control action, including inadequate handling of failures, the Maintenance department might have inadequately actioned the protocol or there may be a missing structure such as cleaning grease might not have been part their protocol.

In the chain of events approach, the next proximate structure would be blamed for the cause of the accident. In this scenario, maintenance department would be blamed for not maintaining the escalator, which would not be an appropriate causal factor.

STAMP analysis has helped in separating the factual data from the assumptions and presents a clear and concise picture of the accident from all the controller's interactions as the analysis is based on identifying the controllers, their interactions, and the control mechanisms as the first phase of the accident analysis. The behavioral dynamics underlying the change in control mechanisms in the next phase identifies the missing structures or constraints. To understand the dynamics of the entire system, by collating the failures at each component that led to the accident is clearly projected in the final phase of the STAMP model.

The control structure models the non-linear relationships reflecting the behavioral dynamics controlling the behavior of entire organizational structure over time[Leveson, 2004]. Thus STAMP is based on a system theory, where components are considered as control structures and each control structure exercises controlling interactions on others, the accident is understood as component failures, dysfunctional interactions among components and environmental disturbances[Leveson, 2004].

STAMP analysis has adopted STPA process in structuring the accident analysis, thereby ensuring consistency in the validation of the model for concurrency where factual data is analyzed by wading through interpreted data, where analysis of contextual behavioral adoptions to change is assessed in the dynamic perspective of the evolution of the constraints, missing control structures are identified, data gathered is thorough from the process perspective as the model encourages identification of data from controller's perspective, constraint's perspective and the missing component's perspective and as the model is encompassing systemic changes in the organization, the politics of inquiries are handled by itself as STAMP has been widely used in software industry.

Thus STAMP is considered to be a formal method of Systemic Accident models. STAMP accident analysis is adopted for this case study as detailed in the chapter 3. In order to understand the wider context, the accident analysis has gathered information about the program context to evaluate the accident context in the wider perspective in the next section.

Chapter II SBIRS program Analysis

2.1 System analysis

SBIRS program analysis in this section will detail the current system which was the resultant of the redesign effort undertaken, in spite of good processes adopted this program was not able to make use of it in attaining higher performance in satellite development and enhancements in various stages of development, test and launch capabilities. Although this program stands as success strategy for other programs to follow, it has been through difficult stages in its life cycle. SBIRS program was the first to adopt new program initiatives, so there were initial problems with effective implementation, this program has overtime successfully molded itself to derive better results from the program initiatives. This program has learned to improve the efficiency and had reduced the cost effectively by inclusive stakeholders, but time had taken its toll. The SBIRS program’s effective processes that were adopted and had taken the experience of the project to improve and excel to break open from the bureaucratic bonds are discussed in detail in this section.

SBIRS program is an Infrared Sensing system which provides sensing ability to detect threats with accuracy and in time. SBIRS program is developed in two systems. The System with Control Segment and User Segment, which provides integration with existing DSP satellites to provide current military capability[SBIRS, 2013]. And the System that includes a space segment consisting of two hosted payloads in HEO and four satellites in geosynchronous orbit. This system provides ground system software and hardware for consolidated data processing across all sensor families[SBIRS, 2013].

The space system with both ground system software and space system has been in the orbit with two HEO payloads and two GEO satellites and additional GEO satellites will be launched in phases over next few years [SBIRS, 2013]. The ground systems software and hardware will be replaced in blocks completing in 2018 as Missile Defense mission performance was improved by integrating GEO-1. Technical intelligence and battlespace awareness missions were functional and effective, as it provided increased revisit rate and more data thereby more efficient in identifying target missile. [SBIRS, 2013].

Following numerous delays caused by software malfunctions and other hardware deficiencies, Airforce had acknowledged that GEO -1 has performed better than expected during trial period and has demonstrated a sensor pointing accuracy “nine times more precise than required” and is capable of “detecting targets 25 percent dimmer than required with 60 percent more accurate intensity measurement than specification”[Evans, 2013].

GEO-1 was launched in 2011, GEO-2 in 2013, joined by GEO-3 which is undergoing testing (acoustic and thermal vacuum), GEO-4 is preparing for final assembly integration and testing, GEO -5 and GEO-6 will follow[Zacks, 2014].

Problems in Program level Initiatives

DOD directives 5000 series were followed to reduce bureaucratic process and procedures[Jay A. Moody, 1997]. Although these promising system concepts were adopted in implementing the chosen architecture the architecture so adopted did prove to be a drastic failure[GAO, 2008]. The system analysis was rigorous to set the initial baseline for system performance requirements and KPP, the basic requirements of the mission were foregone to meet the schedule and cost. SAMP was the document which collated all aspects of the acquisition, this was not appropriate for the military context as more stringent measures were foregone and expressive nature of the various aspects of the program was eliminated for the sake of program’s success banner[Jay A. Moody, 1997].

Phase I Pre-EMD did not define the design concepts to the defined requirement as there were many changes in the engineering design as the program progressed which was ineffective at that stage of adoption[Jay A. Moody, 1997]. Each contractor team prepared systems requirements and systems functional requirements for the pre-EMD but the requirements were not properly defined as it resulted in formulating a high-level architecture which was to be overthrown for its own good[Jay A. Moody, 1997].

COEA was not effectively performed as the cost-effectiveness was derived from the performance analysis which was not performed on the grounds of achieving higher efficiency rather it was based on amicable solution among the Defense organizations[Jay A. Moody, 1997]. USAF requirement was to centralize the processing of satellite data whereas warfighters required the data to be provided directly to the theater combatant commands for processing, but the SPO decided to take a middle ground by centralizing the data processing in favor of warfighters which in the later phases proved difficult to achieve[Jay A. Moody, 1997].

Cost as Independent Variable was developed to maximize the military utility for affordable KPP[Jay A. Moody, 1997], but the affordability ran out of the logical premise as there were many changes to the specifications in battery and power generation aspect of the design. The greatest success of SBIRS characteristics was contractor empowerment[Jay A. Moody, 1997], enhanced communication, reduction in overhead, relaxed documentation and reduced government oversight, although they are considered as success factors, these were the causes of failure, as contractor empowerment led to improper assessment of technical efficiencies which led to cost estimation errors, reduction in overhead led to improper cost estimation as the personnel were not trained in the risk factor analysis, reduced government oversight led to improper coordination between the ground and flight team which together led to unplanned changes in design and many estimation errors in specific technical errors[Jay A. Moody, 1997]. Contractor Logistics support was considered to eliminate military personnel from maintaining SBIRS ground infrastructure which resulted in expensive coordination problems and in effect proved as a wrong decision from the government part to hand over the infrastructure to contractors[Jay A. Moody, 1997].

Problems in Requirements

SBIRS had stable requirements from 1996 to 2005 [GAO, 2003]but the clarity of the requirements was in question at every stage of the life cycle[GAO, 2003]. This led to re-requirement analysis for understanding the operational clarity. De-scope decisions and elimination of unnecessary requirements so deemed at that juncture had managed to prove wrong at the development phase as the workarounds were developed to postpone the inevitable[GAO, 2003]. The workarounds mounted up to a stage where it started its Pareto-optimal tendencies resulting in redesign effort in 2007.

Problems in Architectural phase

CRIMS was developed for technical risk assessment process, but the technical risks were not assessed properly leading to unpredictability in the architecture, development, and test which led to cost overruns[Younossi et al., 2008]. Architecture of the flight software had drastic returns of this failure as the thermal vacuum test in 2003 [Office of the Secretary of Defense, 2005]bore the cost of unpredictability as all telemetry data was lost at halt due to hardware design problems, in development the technologies were not mature enough to (TRL 6) to assess the predictability and in the test the architectural failures had impacted the predictability of what is to come from the development.

The System Evaluation and Estimation of Resources (SEER) were followed to assess the software related technical risks which had similar problems of risk returns[Office of the Secretary of Defense, 2005].

An enabler of progress as stated by Dvorak[Dvorak, 2009], is the problem this project faced in making of flight software which had proven to be a complexity sponge as the design evolved with requirements clarity resulting in problems with quality attributes such as maintainability, testability, interoperability, scalability and flexibility increased multifold beyond repairable stage that in thermal vacuum testing(2007) the problems had increased to the mission failure consequences. As Dvorak advises, "*larger percentage of the resource should be allocated to early analysis and architecture in order to avoid problems and rework later when it is more expensive to fix.*"

Core Flight Software System was developed by GSFC[Mccomas, 2012] and Mission Data System Architecture Platform was developed by JPL[Feller, Gluch, and Woodham, 2010] which in itself is a phenomenal architecture as it did give rise to unexpected performance results in flight software, but as the core flight software stands now is untouchable by their own developers. There seems to be a standing army guarding this core system. It is imperative to note that the core team does not trust others to be part of their honor. The developers were not allowed to change the interfaces and were given specific instructions to develop software based on the restrictive implementation policies.

Problems in Test phase

The testing of ground segment and system testing was not developed in accordance with the military standards by SEIT [Jay A. Moody, 1997], as the test was not ready to accept the development inputs. The test team was still developing the test cases and test architecture was not taken into account before the beginning of the development[Jay A. Moody, 1997].

TEMP was not properly developed as it could not predict the technology risks undertaken by the development team[DOD, 2003]. One single Integrated T&E plan was developed for early validation of software maturity but the process was not utilized to the maximum ability as the test team lacked visibility into the development and risk analysis[DOD, 2003].

Potential failure paths were tested such as Fault Detection, Diagnostics and Recovery (FDDR) for Flight software. Simulated realistic environment “Test as fly and fly as you test” was practiced to avoid failures[DOD, 2003] but the simulation tests had great failures due to workarounds implemented at every stage which mounted up to unavoidable stage[GAO, 2003].

IHC issued interim Authority to operate accreditation after initial operational capability and provided 12 months to gain full accreditation which was inappropriately issued without proper investigation into the assurance capabilities[DOD, 2003].

As understood in this section there were many new initiatives introduced in SBIRS program, but they were not implemented properly leading to an inappropriate analysis of the progress of the program and workarounds were adopted to achieve the assigned goals which did lead to failures in the system. These failures in the process affected the architectural decisions which got to the stage of an accident that will be discussed in chapter three.

The accident is the resultant of the program initiative failures, requirement failures, architectural design failures due to the inadequate clarity of requirements and testing failures due to inadequate development of flight software. As the software progressed through the life cycle, at every stage problems were encountered with workarounds. These workarounds accumulated to the stage of a major failure in thermal vacuum testing in 2007. This had caused a reassessment of the architecture. The resulting architecture was to solve the problems thus far by switching to a single core processor from the multicore processor. This adoption of architecture was the accident discussed in detail.

In the previous chapter, an analysis of methodology to be adapted to perform the accident analysis was discussed. This methodology has to highlight the flaws in the system and bring the actual facts of the accident to light. These methodologies have specific strengths in the analysis of the accident, so a thorough analysis of the accident analysis methodologies was undertaken to understand best-suited methodology to this accident scenario. In the next section how this accident methodology could be applied to the system context of the accident in the case study is discussed.

2.2 Accident Analysis

Accident analysis in the past has problems with the fast pace of technological change, changing nature of accidents, new types of hazards, decreasing tolerance to single accidents, increasing complexity and coupling, more complex relationship between human and automation, changing regulatory and public views of safety[Leveson, 2004]. Accidents with social and organizational factors, system accidents and software errors, Human errors, change processes, holistic approach to the organization and their constraints are basic factors governing the accidents[Leveson, 2004].

STAMP advocates safety must be designed into the system, where development and operations safety has to be controlled using the feedback (measuring channel) and through downward information from hierarchy above (reference channel)[Leveson, 2004]. Feedback is critical to an open system in order to provide adaptive control[Leveson, 2004]. It is noticed that the change is a continuous process and the system adapts to the change. The change process exists in the labyrinth of the context. This context is a guide for equilibrium to be attained between the change and the adaptive processes. This context brings the link between the change processes and the adaption processes of the controls. Nancy Leveson’s theory of reference channel and measuring channel does influence the adaption process, but the context seems to have a disruptive influence on the controls. So it is recommended that context should also be studied for disruptions and prevention techniques should involve context as a factor in their solution.

“*For each of the factors, at any point in the control loop where a human or organization is involved, it will be necessary to evaluate the context in which decisions are made and the behavior-shaping mechanisms (influences) at play in order to understand how and why unsafe decisions have been made.*” [Leveson, 2004]

Leveson's model identifies inadequate enforcement of safety constraints [Leveson, 2004] – *inadequate control algorithm* **-** Nancy explains the context from the perspective of the control whereas there is an overall context which influences not just the control mechanism but the underlying constraints itself which is the most disruptive in course. This underlying context should also be considered as an enhancement over Nancy's model of contextual control. – *Inconsistent process model* - Nancy explains the context form the perspective of the process, the enhancement over Nancy's model would be to include the context which influences the sequence of the process which alters the elements in the sequence leading to disruption. – *Inadequate coordination among decision makers and controllers* - Nancy explains this as boundaries that are trespassed where the context focuses on the individual controller actions, the enhancement over Nancy's model would be to include the high-level context which exists where the assignment of controls are decided. This will help the problem of the context which is set to fail right from the start as contradicting controls would be assigned causing disruption. – *Inadequate Execution of control action* - Nancy explains this as reference channel error, the enhancement over Nancy's model would be to include the execution context as the context which is set for the controllers to behave in a disruptive manner whether the reference channel information is correct or wrong. - *Inadequate or missing feedback* - Nancy explains this as the measuring channel that should have a tap on the state of the context resulting out of the control execution, the enhancement over Nancy's model would be to include the controller context as the context that is most disruptive is the controller itself.

The control structures and control models incorporate the non-linear relationship reflecting the behavioral dynamics controlling the behavior of the entire technical and organizational structure over time in STAMP[Leveson, 2004]. This research has analyzed the STAMP model and enhanced the model to incorporate the context as a separate factor that has to be analyzed independently of the controllers and constraints. The context which formulates the labyrinth for the controllers and constraints to act as defined and adapt to the changes relies on this underlying context. In the above section system context was analyzed and in this section, accident context is derived from the above section for this case study. The STAMP model is enhanced to incorporate the underlying context to perform accident analysis of this case study in the next section.

For example, the software that does not know the plane is on the ground raises the landing gear. Here the context is for the control actions which did not adapt to the change process thereby the context has a disruptive influence on the controller to execute such an action. Nancy's model projects this as an inconsistent process model, which forces the process to erroneously behave by executing the wrong process[Leveson, 2004]. This context which Nancy explains is the execution context which would be the concrete cause, the context which is an enhancement over Nancy's model is the controller context which is expected to have responded to the change process. Thus this research has enhanced the context further to incorporate the wider context to understand the STAMP accident model for its detailed structural understanding of the accident in this case study.

SBIRS program had problems due to the introduction of new initiatives, cost overruns, military procedures were not followed, which led the program to the brink of failure. The program was delayed by 9 years due to many failures. The architecture which was originally in place was proven to be inadequate, the requirements were not followed, the documentation process was reduced, and various other military procedures were reduced to catch up with the program's expectations. This is the wider context set for the accident discussed and this context is delved deep to understand the complexities that were influencing the context.

The requirements were not defined appropriately which led the design to change frequently, requirements were defined without considering all organization's requirements. Initiatives to determine the cost were not taken advantage of, the risk assessment categories were not identified properly which led to the cost estimation problems, the reduction in overhead led to improper co-ordination problems which led to technical errors and design problems. The requirement issues had been managed with workarounds and these workarounds mounted up to create a major failure in thermal vacuum testing in 2007 which mandated redesign.

Technical risk assessments were not done properly which led to unpredictability in architecture. The implemented software was barred from any changes and was restricted from developers. Test architecture was not prepared due to unpredictability in architecture and test did not follow military standards. Assurance procedures were not followed leading to mistrust in the software development.

The original flight software design was to operate on two processors. The flight software underwent testing in thermal vacuum test and had major failures which led to redesign of flight software architecture to operate on a single processor. The accident is the architectural decision taken to operate flight software on the single processor rather than two processors. The inherent problem with this decision was that the flight software architecture would not take the advantage of the multicore processor and distributed application architecture. It is proven in this research that the wider context has a greater influence on the accident. So the wider context is studied to understand the complexities that influenced the architectural decision that this research has conducted accident analysis on.

It is the influence of all the factors in the development life cycle of flight software that had led to this accident. These influences and the pressure mounted due to the program ineffectiveness which led to this accident. The program context that was detailed in the above section and the accident context that was detailed in this section will be analyzed in detail in chapter three. The accident is analyzed using the best-suited method from the chapter one to identify the actual cause of the accident. The accident itself is detailed in chapter five explaining the reason for the architectural decision to be treated as an accident. In chapter four accident prevention process is detailed.

In the next section, STAMP accident methodology is adapted to the case study to understand the accident in the wider program context of the program which has influenced the accident to lead towards the major crisis where a wrong architectural decision was adopted which is the accident analyzed. In the next chapter, the analysis of the case study is done by separating the program context and understanding the problems that influenced the accident and the accident that influenced the program as such. Initially, STAMP methodology is understood in detail and the process to follow in the case study to conduct STAMP accident analysis is understood. The context that had a greater influence on the accident is explained and finally STAMP accident analysis is conducted on the accident.

Chapter III Systems Theoretic Accident Model and Processes

Accidents cause are no more considered as simple due to the growing organizational complexity, they are the result of technological errors, human errors and also historical background and unfavorable organizational context[Qureshi, 2008]. An effective approach to develop an accident model for sociological organizations, the social and organizational causes of the accident is developing traction[Qureshi, 2008].

High technology companies like aviation, maritime, air traffic control, telecommunications, nuclear power plants, defense and aerospace, chemical and petroleum industry, and healthcare and patient safety are complex systems leading to disastrous failures in loss of material and human life[Qureshi, 2008]. Most of the failures are the resultant of organizational factors and human operational factors in technical systems that are part of complex command and control environment. Modern technologies have a significant impact on nature of accidents and would require new causal analysis to understand and to develop prevention techniques[Qureshi, 2008].

STAMP model considers technical, human and organizational factors in complex sociotechnical systems[Qureshi, 2008]. Thus accidents occur due to external disturbances or dysfunctional interaction among system components[Qureshi, 2008]. STAMP is based on the hierarchical model of the sociotechnical system[Qureshi, 2008]. A complex system is dynamic, as it is continually adapting to maintain stability and reacting to internal changes and to disturbances in its environment[Qureshi, 2008]. This system must project safe behavior and show adaptive behavior to cope with the changes[Qureshi, 2008].

Organizations in complex sociotechnical systems with systemic dependencies and tight coupling in the organizational structure and management policies could lead to organizational failures as contributory causal factors in system accidents[Qureshi, 2008]. Organizational context of technological systems is to be considered as it adds to complexity and susceptibility to the system accidents[Qureshi, 2008].

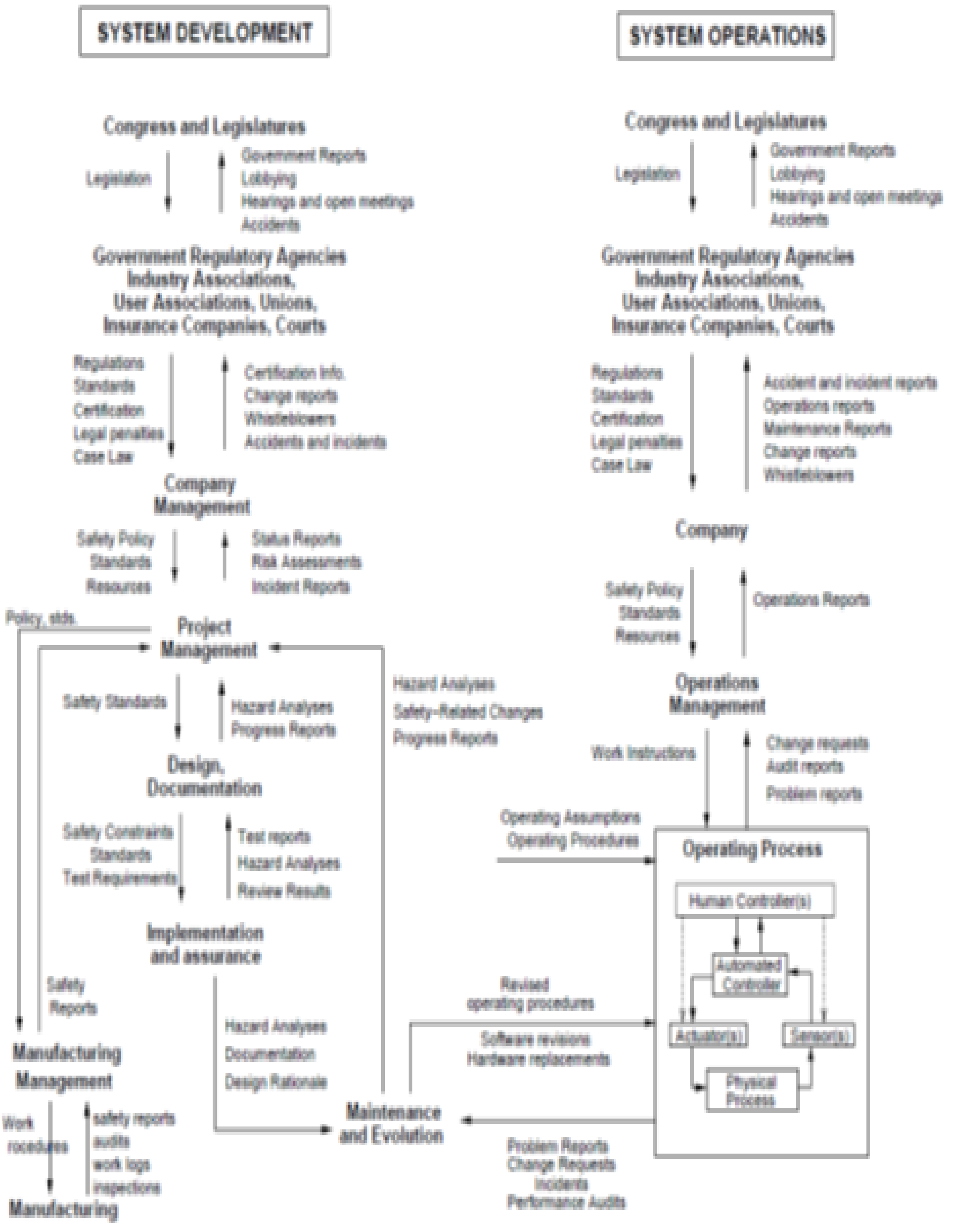
“Vaughn (1996) describes the Challenger accident as "social construction of reality" that allowed the banality of bureaucracy to create a habit of normalizing deviations from safe procedures.”[Qureshi, 2008]

It is essential to understand the role of politics and organizational power which would contribute to accident causation and disasters[Qureshi, 2008]. Resilience engineering is the ability of the organizations to anticipate the changing context to avoid accidents[Qureshi, 2008]. Such system's adaptations cannot be pre-programmed. Thus resilience engineering requires powerful methods, principles, and tools to prevent accidents[Qureshi, 2008]. Systemic accident model is an analytical side of resilience engineering and STAMP has been applied to this case study to analyze the resilience of the organizations confronted by high-performance demands and high risk of accidents.

3.1 STAMP

STAMP is a system theory which analyses process flaws, system component interactions, organizational structures and engineering activities to understand the causes of the accident[Song, 2012]. This model focuses on the operational process of the system design and analysis where *constraints*, *hierarchical levels of control* and *process models* have control flaws leading to accidents. (*Figure 14*)[Leveson et al., 2003]. STAMP explains the processes involved in accidents by analyzing the process model designs where control flows are mapped to hierarchical levels of control to identify the control flaws and gaps in the constraints are analyzed to identify the interaction flaws to prevent accidents in future[Song, 2012].

In system theory, systems are viewed as hierarchical structures where each level imposes constraints on the activity below it as the constraints are the interactions between control structures with details of behavioral structures that help in reaching the goal without failures[Leveson, 2004]. The hierarchical controllers impose these constraints on the interaction context. *Figure 12* shows a generic socio-technical control model with system development and system operation as basic hierarchical control structures and interactions between them[Leveson, 2004]. Between the control structures, interactions are controlled by the information from the controller to enforce the constraints[Leveson, 2004]. *Figure 14* depicts an example of controllers and their interaction between them which has been mapped to the hierarchical levels of the organization[Leveson, 2004].



General Form of a Model of Socio-Technical Control.

Figure 14 STAMP Model [Leveson, 2011]

STAMP describes the accidents as elements in the process acting on the next stage of the process with lack of explicit boundaries and overlapping of authorities between various stages, which generates conflict of interest on the information flow encouraging lack of confidence in the execution of the flow of control leading to accidents when seen as a whole process[Song, 2012].

This model details the intrinsic flaws in the system which hinders in achieving the goal smoothly. An accident here is therefore not just loss of life or property but the deviation in the regulated expectations[Song, 2012]. Thus this research on SBIRS will show the importance of deviation in the expectations during the natural course of continuous process improvement. And the constraints that were imposed on integrating different systems which were dependent on each other for goal achievement. In SBIRS, the systems development and systems operational levels will be detailed in order to understand the deviations which accumulated to build into an accident situation.

The communication between the hierarchical levels are governed by controls as shown in *Figure 15* which constraints the behavior of controls on next level of the hierarchy to avoid accidents[Leveson et al., 2003]. The continuous feedback mechanism improves the process and maintains the process in a dynamic equilibrium. This model is an important concept in STAMP which explains the framework of the process in order to understand the status of the controlled process so that the controller can amend the constraints to have a better understanding of effects on behavior to derive the goal[Leveson et al., 2003]. This model will be enhanced to accommodate other factors that influence the interaction model of STAMP core principles after learning from the SBIRS accident implementation in order to make the model preventive.

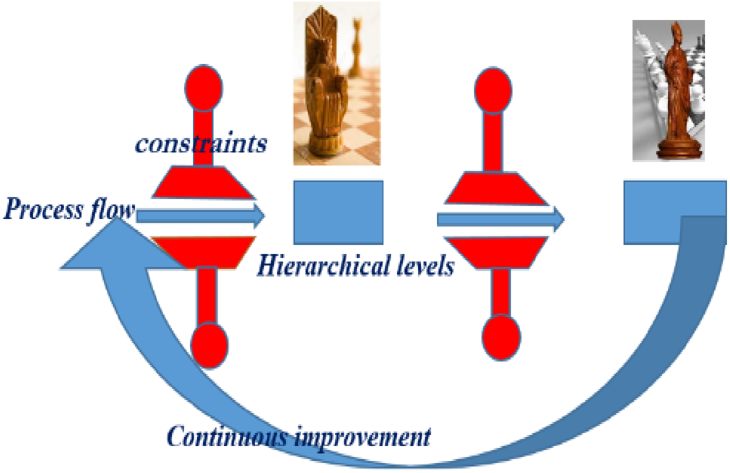


Figure 15 STAMP core principles

STAMP is a model which shows the interaction among the system components with inadequate enforcement of controls or constraints as the cause of the accident. Thus in essence component failures are attributed by the external or internal disturbances resulting in dysfunctional interactions among systems when not handled appropriately by the control system results in an accident[Leveson, 2011].

In SBIRS program various constraints were deviated to speed up the process, the state of behavior was not monitored properly and continuous improvements were adapted simultaneously which resulted in excessive cost and schedule overruns. The architecture of flight software was influenced by the ongoing process of improvement and the architecture had to bear the consequences which resulted in the adoption of the architecture with single core processor, which is the accident that will be discussed in the following sections.

STPA is a process to conduct accident analysis in a systematic procedure. STPA follows STAMP as a methodology to understand the accident by decoupling the processes involved and understanding the deviations in the control structures and then assimilating every component to understand the influence of one another on the entire picture of the accident. The next part of the section will detail the STPA process to conduct STAMP analysis.

3.2 STAMP based Process Analysis (STPA)

STPA is based on the STAMP, where the whole process is viewed as small components and the collection of those components are analyzed together to formulate an opinion on control outcomes to moderate the behavior in order to understand the continuous improvement procedures for effective outcome[Song, 2012].

An accident is a loss of control in the process resulting in process splits leading to miscommunication between the processes which result in unwarranted outcomes[Song, 2012]. After the control failures are identified, constraints are enforced to minimize the system failures.

For E.g.

Accident: *SBIRS project adopted wrong architecture in 2007 as a resultant of many workarounds built over time. As a result, it was delivered with $7.8 billion over estimated cost with schedule delays.*

Safety Constraint: *Software architecture should be designed well in advance to required completion level in order to avoid such cost overruns. The architecture was only 50% complete when development started.*

After identifying the constraint failures the process in the hierarchical control structures should be defined with detailed control outcomes in response to constraints. The general socio-technical model is described in *Figure 14* where the downward lines show the control action and upward show the feedback which provides the effectivity of the constraints to the controllers of the system[Leveson et al., 2003].

The next stage is to identify the inadequate controls which would lead the system to the accident. Thus an accident is a state where the constraints are violated that were already defined in the system. The migration of the constraints is undertaken to prevent accidents. The control flaws are classified as (*Figure 16*):

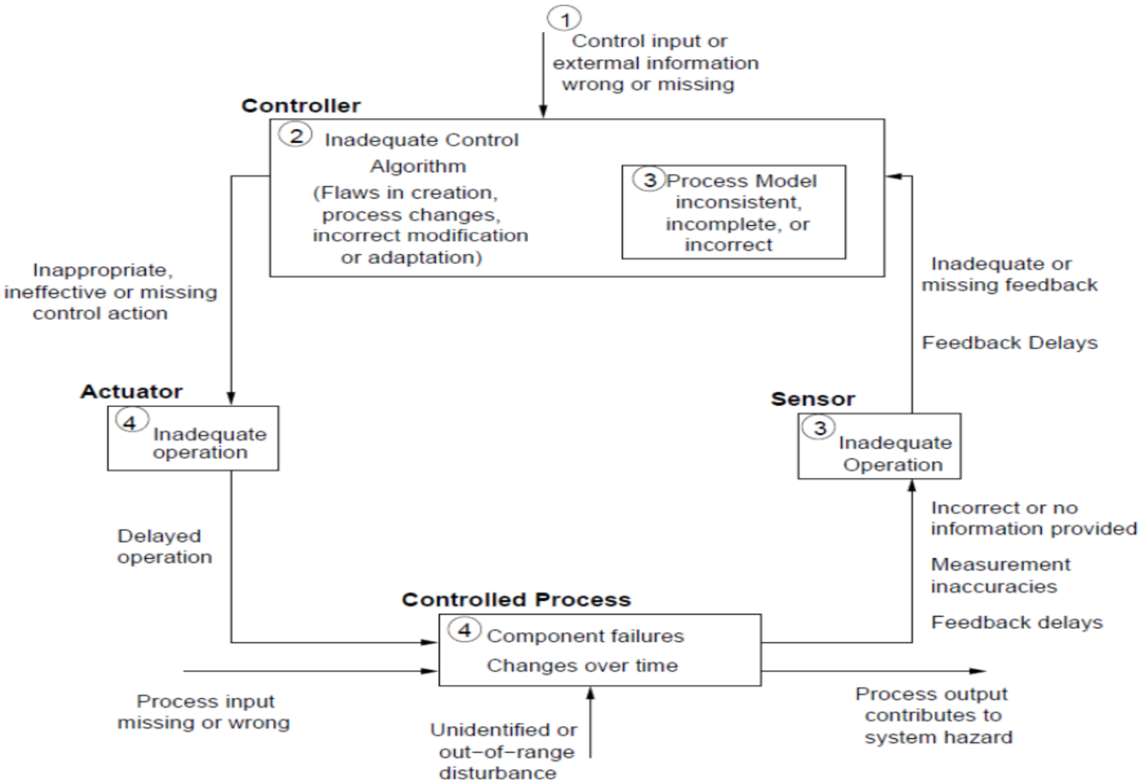

1
Control input or external information wrong or missing
Controller
2
Inadequate Control Algorithm (Flaws in creation, process changes, incorrect modification or adaptation)
3
Process Model inconsistent, incomplete, or incorrect
Inappropriate, ineffective or missing control action
Inadequate or missing feedback
Feedback Delays
Actuator
4
Inadequate operation
Sensor
3
Inadequate Operation
Incorrect or no information provided
Measurement inaccuracies
Feedback delays
Delayed operation
Controlled Process
4
Component failures
Changes over time
Process input missing or wrong
Unidentified or out−of−range disturbance
Process output contributes to system hazard

Figure 16 Classification of control flaws leading to accidents [Leveson, 2011]

1. *Control input or external information is wrong or missing*: control information provided by the controller to controlled process could be missing or wrong.

2. *Inadequate control algorithm:* The control information may be inadequate or due to changes in the system, the information may not be appropriate or the process may be inadequately formed at the beginning.

3. *The process model and Sensor:* The process model may be incorrect from the beginning or changes over time might have made it inconsistent or it might be corrupted due to the inadequate feedback mechanism.

4. *Actuators or controlled process:* The controller may not have adequate process control in place due to transmission failures or it may depend on the input from other system or the component failure.

In addition to above-mentioned control flaws, there could be multiple controllers resulting in communication failures. Thus STPA is a systemic method used in accident analysis as it considers the system as a whole rather than as separate component failures[Song, 2012].

STPA methodology was followed in order to effectively conduct accident analysis of SBIRS program. STPA process was followed in the accident described in the next section. According to STPA methodology, the STAMP accident analysis is implemented in three stages.

1. *The control structures are identified*: At this stage of the accident analysis process, all the control structures are identified, their functionalities are understood (systems development), and their operating boundaries are defined (systems operations) as defined in *Figure 14*.

2. *Changes in control structures are identified that led to the accident*: The next stage is to identify interaction failures among the controllers. The controllers change continually in response to the outcome of the controller's actions (constraints) are understood in accordance with classification criteria as described in *Figure 16*. This is considered as atomic equilibrium achieved between the controllers. This may not consider the whole picture of the organizational changes in other controllers. Those controllers that did not adopt or lost control were considered as inadequate to reach the organizational goal. They are identified as missing structures.

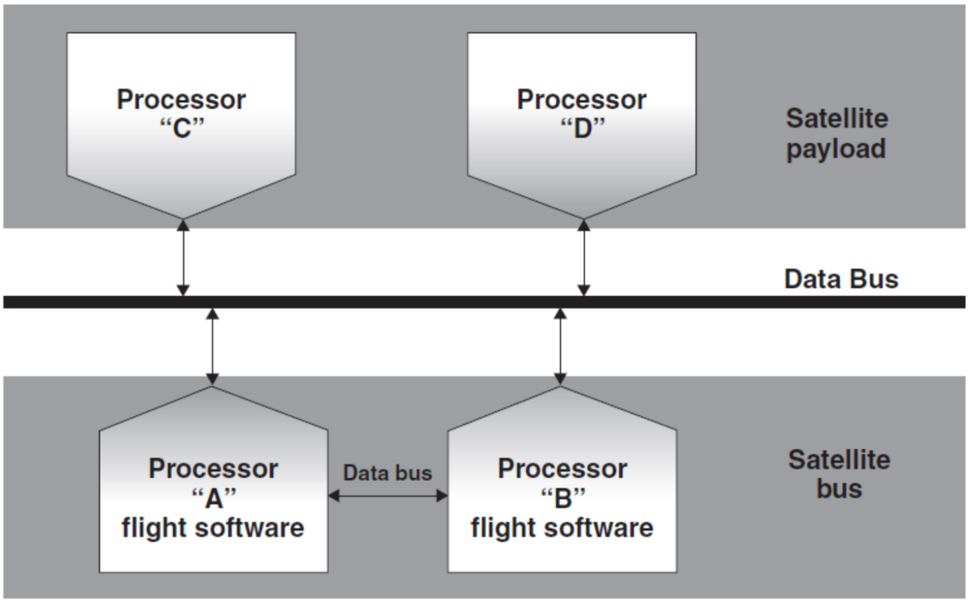
3. *Dynamic mapping of the changes which combined to form an accident*: This stage is to understand how various changes in controllers had influenced collectively towards failure. As one team of the controllers in an organization might not know the difference their decisions make on other organization's controllers. At this stage, a clear picture of the accident cause is understood to derive preventive measures.

The STAMP analysis which helps to identify the controllers and their interactions based on constraints and their responsive change in constraints to stabilize the continuous change is understood clearly as explained in *Figure 15*. STPA process helps to organize the identification process in stages to arrive at the failures in individual processes that lead to accident collectively. The next section explains the situation that led to the accident in SBIRS. The accident is the adoption of the single-core processor for flight software architecture.

3.3 SBIRS Architectural Accident

Flight software controls GEO satellite mission-critical functions such as health, status, and safety like telemetry, thermal control, power management and fault detection activities, so they cannot be deferred and uploaded after launch[GAO, 2008]. The original flight software design was to operate on two of four computer processors onboard the satellite as shown in *Figure 17*[GAO, 2008]. The flight software failed in testing and redesign efforts were planned by simplifying the architecture and increasing the robustness of fault management system for revised cost and schedule[GAO, 2008]. Lockheed Martin conducted trade study and recommended simplified architecture with all applications to be placed in the single processor rather than distributed application as shown in *Figure 18*[GAO, 2008].

Figure 2: Simplified Diagram of Original Flight Software Design



Source: Lockheed Martin (data); GAO (analysis and presentation).

Figure 17 Flight Software Architecture[GAO, 2008]

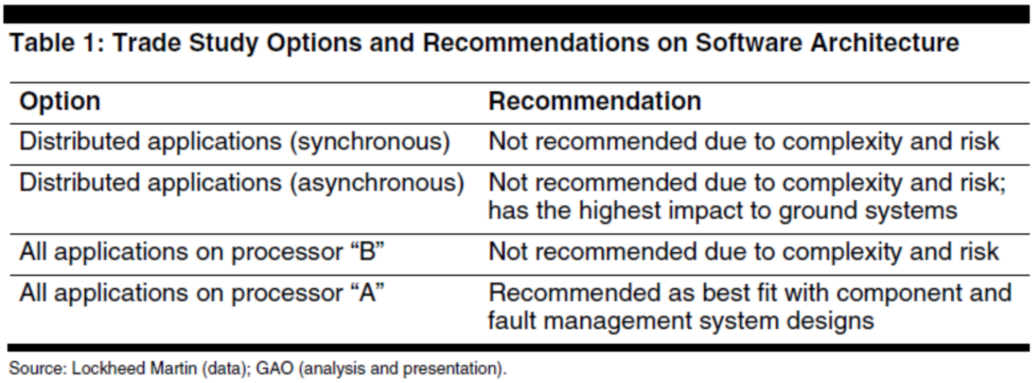


Table 1: Trade Study Options and Recommendations on Software Architecture

Option	Recommendation
Distributed applications (synchronous)	Not recommended due to complexity and risk
Distributed applications (asynchronous)	Not recommended due to complexity and risk; has the highest impact to ground systems
All applications on processor "B"	Not recommended due to complexity and risk
All applications on processor "A"	Recommended as best fit with component and fault management system designs

Source: Lockheed Martin (data); GAO (analysis and presentation).

Figure 18 Recommendations of flight software architecture[GAO, 2008]

The accident was the decision taken by DOD and recommended by Lockheed Martin that all applications were to be placed in one single processor[GAO, 2008]. This accident will be analyzed thoroughly for the circumstance in which such a decision was taken. In this section, the STAMP accident analysis is undertaken on the program level to understand the circumstances leading to the accident.

In 1996, Lockheed Martin started development of reusable flight software (multifunctional bus) and in 2004 SBIRS program adopted flight software for GEO satellite[GAO, 2008]. In 2005 to 2006, Airforce and Lockheed Martin conducted detailed requirements review[GAO, 2008]. In January 2007, flight software underwent thermal vacuum testing and major unexpected and unexplained failures were uncovered[GAO, 2008]. In April 2007, as the defects escalated in additional tests, Lockheed Martin notified DOD of the seriousness of the problem[GAO, 2008].

In April 2007 to July 2007, Airforce and Lockheed Martin developed two options either to modify the existing software or redesign the software by simplifying the architecture. In September 2007 to December 2007, Airforce chose to redesign the software architecture and began detailed software redesign efforts with Lockheed Martin. Lockheed Martin undertook trade study and recommended options as shown in *Figure 18*. Lockheed Martin recommended a simplified architecture to place all applications on a single processor as it represents the best fit with the system design[GAO, 2008]. This design was to address problems in original design such as the timing of stored programs that failed during thermal vacuum tests and fault management system that would increase the robustness[GAO, 2008].

In March 2008, incremental Design Review was conducted for Block 1, which was approved by the program review board for revised cost and schedule[GAO, 2008]. In April 2008, the design was reviewed by six independent review teams for Block 2 design in Systems Engineering & Incremental Design Review which authorized Airforce and Lockheed Martin to proceed with formal software coding under the new design[GAO, 2008].

The accident is the adoption of architecture to place all applications on single processor after the thermal vacuum test failure in 2007. This is an accident as the applications will lose the advantage of distributed application and the choice of the single-core processor over multicore processor is a wrong architecture that was implemented.

This accident was due to the circumstances that forced such an adoption of architecture as a get away from the existing situation. These circumstances are detailed below to understand the influence of the factors that had forced such a decision. The problems in different organizations involved are discussed under which collectively provided an environment for the accident.

Problems in Test

The problems uncovered in thermal vacuum testing were not identified earlier, as test beds were developed in parallel had defects which proved difficult to distinguish between testbeds and flight software issues; oversubscription of testbeds and lack of simulation resources had led to high-risk areas such as timing of stored programs which were not tested for insufficient modelling and lack of robustness[GAO, 2008].

A review was conducted in September 2002 to June 2003 to evaluate development testing of flight software for two ground segments: Highly Elliptical Orbit Intersegment Telemetry, Tracking and Commanding; and Highly Elliptical Orbit Early-On-Orbit Test Mission Processing for completeness, adequacy of testing including planning, execution, and reporting, as a result, validation of security, penetration test of system security features for HEO were found to be incomplete, leading to HEO test data to be doubted and HEO capability to test, assess and support SBIRS was contested[DOD, 2003].

Problems in Development

Ground software development efforts were delayed due to database problems and the total size of software equivalent lines of code which impacted the schedule in system test[GAO, 2007]. In addition software, development and test efforts had integration and total performance problems due to combined SPA and Pointing Control Assembly hardware and software elements and faulty hardware and software design of HEO/GEO flight computers and problems with ‘halt’ anomalies of single board computer[Office of the Secretary of Defense, 2005].

HEO P/L Single Board Computers had problems, there were three occurrences of unexplained P/L anomaly in halt as all the P/L telemetry data was lost and P/L did not respond to commands in thermal vacuum testing of HEO 1 in 2003[Office of the Secretary of Defense, 2005]. As the telemetry data was lost no conclusions could be drawn and this problem repeated in second P/L thermal vacuum testing as well. GEO P/L configuration is different from HEO P/L and complex, latent defects were discovered in the manufacturing process of HEO in integration and test sequence leading to delays in the schedule[Office of the Secretary of Defense, 2005].

As SBIRS High had problems with sensor chip assembly development needed for sensor detector arrays and pointing control assembly software development and control gyro reference assembly also failed during life testing in Increment 1 and ground software problems resulted in two-year slip[GAO, 2003]. As HEO 1 was the first major deliverable for Increment 2, the sensor delivery was delayed by a year from February 2002 to February 2003, further delay was due to first infrared sensor that had significant defects in flight software involving sensor’s ability to maintain earth coverage and track missiles while orbiting the earth in system test in November 2002 resulting in further postponements of delivery[GAO, 2003].

The first time integration of flight software was exercised in a new system, the proto-qualification had to be tested using simulators or flight hardware so transition to facilitate the operational use had added complexity in integration and test for GEO 2 as flight software would be operationally used in development testing which led to remaining SBIRS program at risk[Office of the Secretary of Defense, 2005].

SBIRS High had problems in development integration after restructuring and MR were depleting at a higher rate leading to cost and schedule variance[GAO, 2007]. GAO analysis reported challenges in assembly, integration and test before the re-baseline in February 2006 and schedule delays and cost overruns leading to 28% of MR been spent from April 2006 to November 2006, thus to meet the cost and schedule goals, some needed capabilities were deferred[GAO, 2007].

SBIRS High technology integration was a high risk due to insufficient time as by June 2003 only 58 percent of GEO sensor integration, assembly, test and checkout work was completed with $2million of work were behind schedule[GAO, 2003]. Software development of the ground system efforts was behind schedule by 32 percent by November 2006[GAO, 2007].

Problems in Design

Major design changes occurred to GEO late in design phase due to technical problems found in testing, in 2000, the HEO flattener lens failed during first random vibration test and lens came out of its mounts due to design deficiencies, then the corrector lens failed in the second test in 2001, in addition, due to degraded sensor performance which if left unaddressed would lead to failure to meet KPP, which was resolved by adding 12 foot sunshade for off-axis solar radiation rejection. HEO 1 had continuous changes to design due to SPO authorizing to pass the SBIRS High critical design review with just 50% of design drawing whereas the recommended completion was 90%, in addition, IRT report found that the program did not invest enough time and resources in the basic systems engineering analysis resulting in cost and schedule escalations[Younossi et al., 2008].

Two late design changes were made to improve GEO satellites success, 80 amp battery was to be replaced with 100 amps battery to improve operational reliability with the estimate of $15 million but the cost performance report in June 2003 shows the contractor was having difficulty assessing the specifications of the battery resulting in schedule delays and increased cost. And the second change was to modify the solar cell panel to resolve power deficiency for which the impact on cost was not yet determined[GAO, 2003].

Problems in Program

The SPO had limited ability to identify all technical risks due to lack of integrated management system as contractors had projected optimistic claims on work content completed resulting in inaccurate and unrealistic cost performance index and schedule performance index, up to date information was not available for thorough technical assessments and interrelationships among risks were not identified, and IBRs were not conducted regularly, on top of that SPO's visibility was limited as consequence of TSPR[1] which removed the level of rigor in monitoring and assessing contractor capabilities, technical assessments were subjective consequently the inexperience of the staff had influenced, budget pressure, rushed environment and optimism about TSPR had influenced technical risk assessments[Office of the Secretary of Defense, 2005]. HEO and GEO payload development had series of technical issues in 2002 but in 2002 IBR risks were rated moderate or lower, it is unclear why it was evaluated in a more optimistic light. The contractor risk assessments were either incomplete or over-optimistic due to contractor's own incentive to bias the technical assessments[Office of the Secretary of Defense, 2005].

TSPR approach had poor performance which was reacquired in 2002 to regain the ability to properly oversee and assess contractor performance by SPO[GAO, 2008]. COTS were assumed to be beneficial and hence deemed low risk without in-depth analysis as it was based on commercial bus, but GEO bus underwent significant configuration changes and weight growth due to unique military requirements consequently led to removal of military standards and specifications which in turn led to quality control issues that resulted in serious technical problems such as HEO EMI problem[Office of the Secretary of Defense, 2005].

[1] ***TSPR – Contractor formulates technical design, implements solutions and relieved of cumbersome reporting requirements with minimal government oversight.***

Contractor oversight was not appropriately managed which led to lack of disciplined approach to software development resulting in inadequate coordination between cost and schedule functions[GAO, 2008]. Further to meet the cost and schedule goals, waivers were granted by software engineering process group to software development process which increased the program risks[GAO, 2008]. Waivers were approved for software design to be done in parallel with software specification activity, which led to certain requirements been rejected and resulting in rework in design and coding. Another waiver was authorized for software unit integration testing to be done in parallel with formal unit testing, as a result, formal unit testing found problems that were not found in development unit testing resulting in rework[GAO, 2008].

SBIRS High maintained insufficient memory margin of the onboard satellite of 35% as against 50 % required and waivers were granted for this effect[GAO, 2003]. Thus SBIRS program had continued technical complexity challenges leading to flight software failure in thermal vacuum testing in spite of more than 12 years of its inception which had resulted in cost overruns[GAO, 2008]. There were problems with acquisition policies governing basic system design which led to latent defect and process escapes in AI&T activities resulting in cost and schedule delays due to rework in GEO[Office of the Secretary of Defense, 2005].

Technical risk assessments were underestimated due to immature technologies, compressed testing schedules which led to technical difficulties that eventually resulted in failures in meeting technical performance which in turn led to redesigns and reworks leading to schedule slips and cost growths[Office of the Secretary of Defense, 2005]. In 1996 risk assessment, all other risks were mitigated except HEO software development activities continued as a risk in GEO integration and test and in 1999 risk assessment, again flight software did not receive much attention and was found to be unexpectedly difficult in HEO integration. In 2004, technical risks related to flight and ground software related to first-time integration efforts were found to be of high risk[Office of the Secretary of Defense, 2005].

In addition DCMA reported variance at completion at $25.6 million and schedule variance of 32% whereas threshold variance was 5% due to integration testing and operations, thermal vacuum test preparation and engineering rework such as Pointing and Control Assembly software was restructured to allow off-ramp option whereas flight software could not be off-ramped as they are needed for launch, tracking algorithms and software were not complete with hundreds of open defects and delayed qualifications[GAO, 2007].

In addition lack of coordination between ground software development and space due to late delivery of database and inability of program office to reduce the length of time taken to certify data processed from GEO1 resulted in accelerated ground software development. In addition integration of GEO flight software which was high-risk effort did not start until August 2003 as scheduled[GAO, 2007].

There were 148 defective EMI frequencies, which would lead to mission failure, of which 39 design modifications were made and 7 were granted waivers as it would not cause performance problems. There were process failures, stringent requirements and the subcontractor did not implement the EMI control plan which was agreed in EMI design review and further no contractor was clearly responsible for integrating HEO sensor with the host bus[GAO, 2003]. Further due to signal weaknesses in HEO sensor, which had delayed and increased the risk. SBIRS continued to experience technical issues in GEO signal processing software development and HEO-2 payload software qualification testing after HEO sensor delivery[GAO, 2003].

Hardware installation at the remote ground station, legacy reporting system interfacing with MCS, delays to start testing, the requirement of testing of parallel operations prior to the declaration of IOC, fault detection, and isolation problems led to a significant delay in performance and reliability test. And HEO message certification milestone schedule had delayed from November 2004[DOD, 2003].

Compressed timeline, issues due to shared facilities at overseas relay ground stations, delay in performance validation testing of increment 1 ground software, inadequate testbed design and scope, simulation tests needed, accelerated deployment of low component in 2004, significant improvement in SBIRS High requirements over DSP and inadequate HWIL testbeds in OT&E has resulted in increased risk and schedule delays and cost overruns[DOD, 2003].

The decline in the defense budget, consolidation of the aerospace industry and growing competition for the fewer programs, resulting in increased cost efficiency by transferring program responsibilities to contractors with less government oversight[Younossi et al., 2008]. The acquisition reform measures eliminated usual cost and technical risk assessment data recording, increased technological complexity and reduced acquisition workforce due to downsizing challenged the knowledge to assess the technical and system engineering progress of the program[Younossi et al., 2008].

In addition the program office did not implement Airforce instruction 99-101 “Developmental Test and Evaluation” also SMM tool was not used to track the progress of the program and all critical test plans and reports were not signed off, thus without effective management and oversight of development and testing, the program had the risk of repeating the problems identified during the program recertification[DOD, 2003].

The SBIRS program had a major failure in thermal vacuum testing in 2007, which was the consequence of the problems found in testing, development, design, and program. The thermal vacuum test failure was not detected earlier as testing was not performed adequately, the initiatives which were introduced did not allow thorough testing to be performed. The development problems were mostly due to new technologies and integration problems. The thermal vacuum testing problem in 2003 was not clarified when the same problem repeated in 2007 with additional problems due to workarounds done to cover up the minor problems mounted to an unmanageable extent. The development did have problems in design issues where they had to make up with workarounds to cover up the design issues. The design problems were handled in development and were caught in the test. Thus the development did bear the problems from the design and test. The program initiatives were not giving fruitful results, the problems with contractor oversight, waivers were granted, the disciplined approach was not adopted, the design was not completed before starting the development, pressured by the political situation of the program to finish were the major factors that had an influence on the program.

The scene of SBIRS program was set with many problems from various departments. These problems (mentioned above) were derived from System Analysis done *in Chapter II SBIRS program Analysis.* The program underwent changes from controllers of other organizations and within to incorporate the continuous change by molding the constraints to establish a temporary equilibrium. In doing so, many of the organizations did not cope well to achieve the goal.

These above-mentioned failures had resulted in the loss of resources, money and time. Accidents should be analyzed from the point of failure in achieving efficiency, not just the end result of the project.

Although the SBIRS program did manage to complete the project successfully, it had major failures to learn from the cost overruns, program delays and initiatives introduced in the processes. This time it just caught up with the thermal vacuum test in 2007 which forced an action, this architectural decision was a wrong decision taken which would further degrade the development.

As the problems were identified from different organizations (above mentioned), the next stage in accident analysis process is to identify the controllers, functionalities, and constraints they have control over. How these above-mentioned changes had brought forth equilibrium among the interacting controllers will be analyzed in the next section.

3.4 STAMP analysis of SBIRS Architectural Accident

SBIRS program had major defects that mounted up to the mission-critical problem in thermal vacuum test which resulted in $7.8 billion cost overrun and schedule delays of 9 years. An accident analysis is performed to understand the causes of the accident and to learn lessons from it[Younossi et al., 2008]. STAMP analysis is conducted to identify the flaws in decision processes and control flaws that led to the accident.

SBIRS program sets the stage for larger context over the years for decision failures overtime which mounted up to be a major problem. The decisions so taken at that juncture were not noticeably failure control factors, but over time as the process evolved, even the slight deviation formulated a major failure. This evolution of failure control factors could be in different parts of the system that will not know how it will affect the other parts of the system. This could even be a result of just one part of the system that had degraded resulting in a domino effect.

The first step in creating STAMP analysis is to identify system failures (identified in the previous section) and system constraints. Each part of the process will have system constraints, these safety constraints will have to be adequately designed for the overall system to be effective. This research has taken the additional step of identifying the overall context and how it influences the accident context in the previous section.

The accident is shown in two phases, one as a static snapshot of control structure over time for many such snapshots to formulate series of control structures which does not show dynamic nature of the accidents. The next phase is to depict the dynamic nature of the model showing the relationship between control structures and resulting failure events. The final model summarizes the other models and depicts the accident causes evidently by showing for each control structure, its decisions, and control factors that led to an accident so that preventive measures could be recommended.

3.4.1 SBIRS Control Structures

The architectural accident of SBIRS flight software is discussed in detail, the decision of placing flight software in the single processor after the redesign is analyzed for changes in a control structure that led to accident situation. A complete analysis of the program had to be done while considering circumstances in which this decision was taken. The safety control structure should not have decided to place the flight software component in one processor:

- *As single-core processor has lower performance, is less efficient, has lower fail-safe and produces more heat.[Ghuman, 2016]*
- *Application on single core processor is difficult to maintain an application grows in size increasing complexity.[Fielding, 2000]*
- *DOD should not have authorized the design with single core processor.*

The *Figure 19* shows the control structures identified. As the decision was to adopt single core processor and to place all applications in single core processor, there are many problems related to such a decision. The single core processor has a lower fail-safe mechanism and produces more heat. The original design was to have a multicore processor, which is a good architectural decision. The applications on non-distributed application architecture would over time grow in complexity and would be difficult to maintain. Thus the accident is an architectural decision taken without considering these drawbacks. There should have been safety constraints to monitor the architectural decisions. These constraints should have been enforced by DOD in the entire control structure.

GAO is responsible for auditing, evaluation, and recommendation of options to the federal government to make an informed decision. GAO chose to opt for redesign option so presented after the trade study with single core processor. It is the responsibility of DOD to ensure the constraints are in place to effectively design flight software architecture for efficiently performing software. DOD did not have a proper mechanism to evaluate the redesign options presented by Lockheed Martin.

The control structures are identified in *Figure 19* and their functionalities and constraints are analyzed to identify the failures in response to the change (*inference explained in italics*). Controllers like DOD, GAO, and constraints such as waivers are identified and depicted in *Figure 19* and explained below.

Figure 19, explains the relationship between each controller and identifies the failures in interactions. The controller Lockheed Martin which identified major issues of “flight software problems” notified the DOD of its concerns. As GAO is an auditing body, the GAO conducted “performance audit to provide sufficient evidence” of the situation with Airforce, Lockheed Martin, Defense contract management agency and forwarded its recommendations as “assessment of flight software problem” to DOD. The Airforce “recommended the redesign” to DOD. The Program review board “approved the redesign” after reviewing the redesign to DOD. Independent review team comprising of Aerospace Corporation; Lockheed Martin and Under Secretary of Defense “authorized to develop on redesign” to DOD. The Joint Execution team comprising of Airforce, Lockheed Martin and Aerospace Corporation “authorized to develop on redesign” to DOD. Defense contract management agency had submitted “program assessment report” to SPO. USSPACECOM which was responsible for “requirement issue” reported to DOD. Inspector General of DOD was responsible for test operations “reviewed software testing” on Lockheed Martin, Airforce and Northrop Grumman and reported to DOD. Designated approval authority approved “interim authority to operate HEO capability” to SPO. Department of Defense Information Security Certification Authority reported “security validation” problems to DOD. System engineering and integration team “maintains traceability of HEO requirements in test” reported to DOD. Integrated product team, “maintains traceability of ground requirements in test” reported to DOD. The Air Force Operational Test and Evaluation Center reported “operational testing” report to DOD. Space technical interchange had the responsibility of “flight software test plan” reported to DOD. Failure review board maintained “anomaly management documentation” for DOD. SPO conducted “risk assessment” and reported to DOD. DOD authorized “waivers” to SPO. Software engineering process group requested waivers to DOD.

The detailed controllers and their interactions with other controllers are explained below under each controller identified in *Figure 19*. The first stage of STAMP analysis is to identify control structures responsible for the accident.

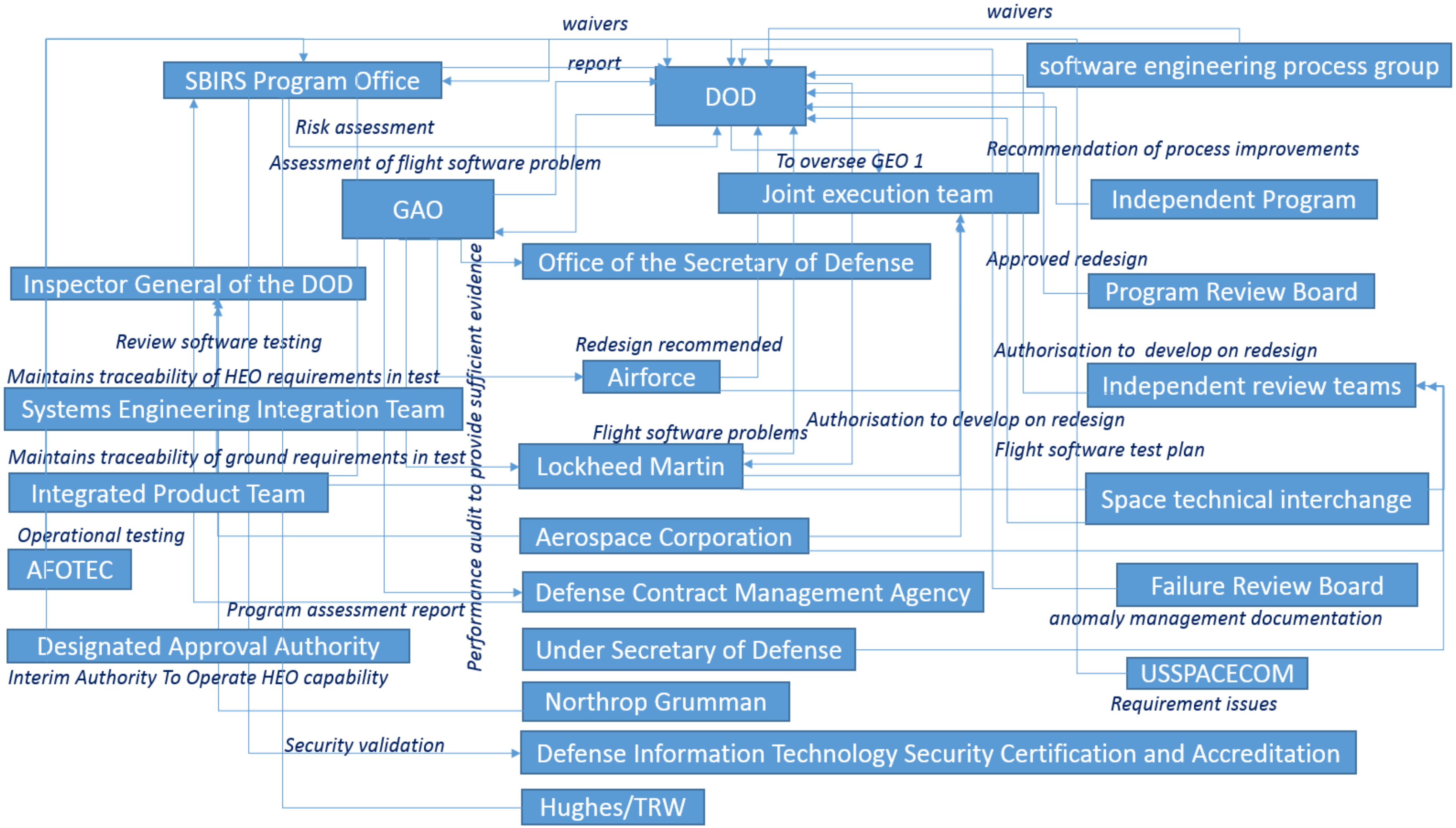

waivers
waivers
report
SBIRS Program Office
DOD
software engineering process group
Risk assessment
Assessment of flight software problem
To oversee GEO 1
Recommendation of process improvements
GAO
Joint execution team
Independent Program
Office of the Secretary of Defense
Approved redesign
Inspector General of the DOD
Program Review Board
Performance audit to provide sufficient evidence
Review software testing
Redesign recommended
Authorisation to develop on redesign
Maintains traceability of HEO requirements in test
Airforce
Independent review teams
Systems Engineering Integration Team
Flight software problems
Authorisation to develop on redesign
Flight software test plan
Maintains traceability of ground requirements in test
Lockheed Martin
Integrated Product Team
Space technical interchange
Operational testing
Aerospace Corporation
AFOTEC
Failure Review Board
Defense Contract Management Agency
Program assessment report
anomaly management documentation
Designated Approval Authority
Under Secretary of Defense
Interim Authority To Operate HEO capability
USSPACECOM
Northrop Grumman
Requirement issues
Security validation
Defense Information Technology Security Certification and Accreditation
Hughes/TRW

Figure 19 STAMP SBIRS Control Structures

Lockheed Martin is the prime contractor responsible for the development of flight software. Since Lockheed Martin was having a major failure in thermal vacuum testing and the defects were growing to an uncontrollable extent, Lockheed Martin proposed an easy solution to solve the problems. The solution was to redesign the architecture with single core processor and to place all applications in the single processor. *Lockheed Martin did not evaluate the consequences of such a design option.*

Government Accountability Office conducted audit in Office of the secretary of state, Airforce, Lockheed Martin and Defense Contract Management Agency to provide sufficient evidence of the problem. GAO did have oversight responsibility on Lockheed Martin during the trade study, still, GAO recommended this option of redesign to DOD[GAO, 2008]. *GAO did not have proper oversight on the contractor's situation and had taken Lockheed Martin's proposed option to be apt.*

Airforce was working with the contractor in the redesign effort and had visibility of requirements of the user community. Airforce undertook requirements re-clarification effort along with the Lockheed Martin just before the redesign effort, still, Airforce recommended this redesign option[GAO, 2008]. *Airforce's basis to recommend this redesign option is not clear.*

Program review board has the responsibility of assessing the architecture to the suitability of military requirements and recommends the design option after the review of the program members[GAO, 2008]. *It is due to the negligence on the part of the review team to have recommended the redesign option so presented by Lockheed Martin.*

Independent Review Team comprising of Under Secretary of Defense for Acquisition, Technology, and Logistics; Aerospace Corporation; Lockheed Martin; Airforce Space and Missiles System Center Wing; and Software Engineering Institute was responsible to give an independent assessment of design, as this redesign effort is followed by major failure in thermal vacuum testing due to design problems[GAO, 2008]. *Independent Review Team did not do a rigorous analysis of the design presented for a redesign due to oversight of the future performance problems this design would arise.*

Joint Execution Team is a joint effort by Airforce, Lockheed Martin and Aerospace Corporation have the responsibility of flight software development of GEO1 efforts and to conduct inch stone review, Executive Program management and to address weaknesses including Independent Program Assessment recommendations on technical baselines[GAO, 2008]. The IPA finding reported on lack of disciplined process of Lockheed Martin and Airforce had limited control on SBIRS program and recommended separate program manager for flight software team[GAO, 2008]. This team was formed after the DOD authorized to proceed with development based on the redesign, it should be noted that the members were already players in the program. This effort was initiated by DOD to mitigate problems related to the original design of flight software. *It is indeed noted that this new combination did not help improve the situation in GEO1 flight software architecture as the development proceeded after the redesign based on the flawed architecture.*

Defense Contract Management Agency has the responsibility of monitoring the progress of software development of GEO in Lockheed Martin as any delays would affect the launch[GAO, 2008]. HEO software development was delayed due to an aggressive schedule and lack of understanding of the complexity of software tasks resulting in higher defects. Flight software sensor ability to maintain earth coverage and track missile while orbiting the earth had several defects in testing and HEO software development is among top ten program risks[Office of the Secretary of Defense, 2005]. In addition, flight software development was significantly behind schedule (32%) and rework due to a higher amount of growing defects. Software development and integration, testing and assembly had problems including a sensor, pointing, and control assembly[Office of the Secretary of Defense, 2005]. *In effect, DCMA did not have a close monitor on the development schedule and the progress on Lockheed Martin resulting in unexpected outcomes.*

Inspector General of Department Of Defense has the responsibility of reviewing software testing and Quality Integrity Accountability testing in development in Lockheed Martin, Airforce and Northrop Grumman[DOD, 2003]. Validation of system security features in Interim HEO capability found that the accreditation process was incomplete[DOD, 2003]. *Thus the data which were tested in HEO capability could not be trusted and the facility to perform tests were also questionable.*

USSPACECOM produces SBIRS CONOPS, SBIRS ORD, JROC validated MNS, SBIRS CRD, and SBIRS ORD. JROC also focused on survivability and data availability for Pre-EMD. Along with SWF and Air Force Requirements Oversight Council served to resolve operational requirements issues[Jay A. Moody, 1997]. The survivability requirement is regarding maintaining nuclear survivability during the cold war and data availability is regarding providing unprocessed data to the warfighter in theater combatant commands for processing[Jay A. Moody, 1997]. *Although the requirements were well captured, the clarity of certain operational details required was not detailed well. So the requirement in the form of clarity kept improving which brought unplanned changes in development which led to increased complexity leading to growing defects.*

Designated Approval Authority is responsible for certifying the security test and evaluation and penetration test, to accredit, withhold or issue Interim Authority to Operate. Designated Approval authority had inappropriately issued Interim Authority to Operate to Interim Highly Elliptical Orbit Capability (IHC) and this is planned to continue till 2010[DOD, 2003]. This violated **Department of Defense Information Security Certification Authority** by not ensuring the system security features were met by conducting security tests. And to allow IHC to operate incorrectly issued Interim Authority to operate annually, System Security Authorization Agreement was violated. *Thus the IHC data is considered incorrect as system security features such as availability, integrity, authentication, confidentiality were not validated against.*

System Engineering Integration Team maintains HEO specification to ground segment requirement using Modified Design Compliance Matrix which is a requirement verification Ledger which is used in testing[DOD, 2003]. The ground segment **Integrated Product Team** maintains software requirements specification of the ground segment which uses Requirement Traceability and Management tool for test verification. These testing results are analyzed to proceed to integration and system testing[DOD, 2003]. *The system engineering integration team did not validate the tests properly to proceed to system test as the code reached the thermal vacuum testing and had major failures.*

Space technical interchange, a testing plan was recommended for space vehicle testing including for flight software[GAO, 2008]. DOD conducted the space technical interchange meeting to improve on the testing capabilities after the redesign effort was approved.

The Air Force Operational Test and Evaluation Center (AFOTEC) has the responsibility of performing Operational Utility Evaluation of SBIRS. Testing of ground architecture, GEO 1, two hosted infrared payloads in HEO and legacy Defense Support Program assets[Office of the Secretary of Defense, 2005]. Joint Interoperability Test Center, OSD, and SPO were brought together with AFOTEC to work as a team and formulate a unified development test and evaluation and operational test and evaluation to formulate a single test and evaluation plan. AFOTEC also ensures the acquisition strategy is maintained throughout the acquisition cycle by validating the operational effectiveness and checks the suitability of system in a cost-effective manner[DOD, 2003]. *Although test had unified approach to development and operational test, the testing was not performed effectively to avoid major failure in thermal vacuum testing.*

Failure Review Board has the responsibility of analyzing the failures and ensuring a mitigation plan for the assessed functionality[Office of the Secretary of Defense, 2005]. *The failure review board had assessed some functionalities as risky, the mitigation plans did not work leading to the risks carried further into later stages of the development cycle. This led the risks to be escalated as bigger to the stage of failure.*

SBIRS Program Office has the responsibility of running the program smoothly to the effective delivery. The program office had brought many new initiatives to have the feel of success, *unfortunately, there were many failures leading to major failure in the delivery of flight software in thermal vacuum testing.*

Waivers were requested by Lockheed Martin in software development process to bypass the regular process to **software engineering process group** which was granted by SPO leading to failures in development process[GAO, 2008]. Two major failures were noted due to waivers, waivers were granted for design to be done in parallel with specification activities and waivers were granted for development testing to be in parallel with formal unit testing. *These waivers evidently led to problems in design as there were many design changes resulting from requirement specification clarification activity. And there were mounting defects in the formal unit testing due to the code not been tested in development testing resulting in heavy rework.*

The next stage in STAMP analysis to identify the changes in control structure leading to the accident.

3.4.2 Changes in Control structure leading up to the Accident

SBIRS program structure started with control problems, over time many new initiatives were brought in to mitigate the control problems, but the problems multiplied with program's progress. There were many deviations in the process to attain the success of new initiatives. Thus small changes in the process or the initiatives could lead to failure.

The failures of the individual controllers from the previous section (3.4.1) are collated based on the functionalities that affected the constraints in response to the changes. The failures are classified based on classification categories (see *Figure 16*) as specified in STAMP (*inferred in italics*) below. As a result, STAMP helps in identifying the controllers that did not adequately respond to the changes which resulted in an accident. The controllers are analyzed for every interaction between them and interaction failures are identified and categorized based on STAMP to reveal the flawed controllers depicted (in dotted lines) in *Figure 20.*

SBIRS Program Office

SBIRS program office has implemented all the initiatives as suggested by DOD and has achieved a certain level of submissive acceptance of DOD's procedural compression of enthusiasm to successfully complete the program. It is evident from below mentioned evolution of attitude by training obedience to follow the set path of DOD.

DCMA reported variance at completion at $25.6 million and schedule variance of 32% whereas threshold variance was 5% due to the integration testing and operations, thermal vacuum test preparation and engineering rework such as Pointing and Control Assembly software was restructured to allow off-ramp option whereas flight software could not be off-ramped as they are needed for launch, tracking algorithms and software were not complete and hundreds of open defects and delayed qualifications[GAO, 2007].

In addition lack of coordination between ground software development and space due to late delivery of database and inability of program office to reduce the length of time taken to certify data processed from GEO1 resulted in accelerated ground software development. In addition integration of GEO flight software which was high-risk effort did not start until August 2003 as scheduled[GAO, 2007]. (***Missing control structure***). *This is because the assumption of estimation did not include clarity of the basis for assessment which led to unpredictable outcomes.*

TSPR approach had poor performance which was reacquired in 2002 to regain the ability to properly oversee and assess contractor performance by SPO[GAO, 2008]. COTS were assumed to be beneficial and hence deemed low risk without in-depth analysis as it was based on commercial bus, but GEO bus underwent significant configuration changes and weight growth due to unique military requirement consequently led to removal of military standards and specifications which in turn led to quality control issues that resulted in serious technical problems such as HEO EMI problem[Office of the Secretary of Defense, 2005]. (***Inadequate control enforcement***). *This is because of negligence in assessing the requirements.*

The SPO had limited ability to identify all technical risks due to lack of integrated management system as contractors had projected optimistic claims on work content completed resulting in inaccurate and unrealistic cost performance index and schedule performance index, up to date information was not available for thorough technical assessments and interrelationships among risks were not identified, and IBRs were not conducted regularly, on top of that SPO's visibility was limited as consequence of TSPR (*TSPR – Contractor formulates technical design, implements solutions and relieved of cumbersome reporting requirements with minimal government oversight.*) (which removed the level of rigor in monitoring and assessing contractor capabilities.[Office of the Secretary of Defense, 2005](***Inadequate control structure***). *This is because of inadequate information about the progress in many systems.*

Technical assessments were subjective consequently the inexperience of the staff had influenced, budget pressure, rushed environment and optimism about TSPR had influenced technical risk assessments**.** HEO and GEO payload development had series of technical issues in 2002 but in 2002 IBR risks were rated moderate or lower, it is unclear why it was evaluated in a more optimistic light. The contractor risk assessments were either incomplete or over-optimistic due to contractor's own incentive to bias the technical assessments[Office of the Secretary of Defense, 2005]. (***Inadequate control structure***). *This is because although technical risks were identified the level of risks were inappropriate, they were not mitigated before the beginning of the development which led to failures.*

Technical risks assessments were underestimated due to immature technologies, compressed testing schedules which led to technical difficulties that eventually resulted in failures in meeting technical performance which in turn led to redesigns and reworks leading to schedule slips and cost growths[Office of the Secretary of Defense, 2005].(***Asynchronous evolution***). *This is because of the complexity of the technology.*

In 1996 risk assessment, all other risks were mitigated except HEO software development activities continued as the risk in GEO integration and test and in 1999 risk assessment, again flight software did not receive much attention and was found to be unexpectedly difficult in HEO integration. In 2004, technical risks of flight and ground software related to first-time integration efforts were found to be of high risk[Office of the Secretary of Defense, 2005]. (***Inadequate control structure***). *This is because the barriers to proceed were not defined.*

Contractor oversight was not appropriately managed which led to lack of disciplined approach to software development resulting in inadequate coordination between cost and schedule functions. Further to meet the cost and schedule goals, waivers were granted by **Software Engineering Process Group** to software development process which increased the program risks[GAO, 2008]. (***Asynchronous evolution***). *This is because of negligence to adhere to the process leading to problems in assessment processes.*

SBIRS High had problems in development integration after restructuring and MR were depleting at a higher rate leading to cost and schedule variance. GAO analysis reported challenges in assembly, integration and test before the re-baseline in February 2006 and schedule delays and cost overruns leading to 28% of MR been spent from April 2006 to November 2006, thus to meet the cost and schedule goals, some needed capabilities were deferred[GAO, 2007]. (***Inadequate control structure***). *This is because the progress of the process had many failures in the delivery mechanism.*

Waivers were approved for software design to be done in parallel with software specification activity, which led to certain requirements been rejected and rework in design and coding. Another waiver was authorized for software development unit testing to be done in parallel with formal unit testing, as a result, formal unit testing found problems that were not found in development unit testing resulting in rework[GAO, 2008]. SBIRS High maintained insufficient memory margin of the onboard satellite of 35% as against 50 % required and waivers were granted for this effect.

Thus SBIRS program had continued technical complexity challenges leading to flight software failure in thermal vacuum testing in spite of more than 12 years of its inception which had resulted in cost overruns[GAO, 2008]. There were problems with acquisition policies governing basic system design which led to latent defects and process escapes in Assembly &Integration Testing activities resulting in cost and schedule delays due to rework in GEO[GAO, 2008]. (***Inadequate control enforcement***). *This is because of the lack of understanding of the extent to which change will cause to the mission goal.*

SPO had performed diligently by following DOD in keeping pace with its directions. SPO had conducted technical assessments but was not able to assess the effects of minor deviations resulting in the projection of compliance to the cohesive environment of DOD. TSPR being a flagship initiative of DOD, did not allow appropriate oversight leading to in- appropriation built in which is later expressed as inertia. Waivers were granted by DOD to bypass some of the norms which led SPO to be submissive in compliance of DOD’s procedures.

AIRFORCE

Hardware installation at the remote ground station, legacy reporting system interfacing with MCS, delays to start testing, the requirement to test parallel operations prior to the declaration of IOC, fault detection and isolation problems led to a significant delay in performance and reliability test. And HEO message certification milestone schedule had delayed from November 2004[DOD, 2003]. (***Inadequate control enforcement***). *This is because of the lack of motivation to improve the efficiency of the process performance.*

Airforce did not enforce compliance at different phases of the development, test processes which had led to a relaxation of grip over the contractors. So the contractors had responded to this attitude by keeping Airforce in dark.

LOCKHEED MARTIN

Lockheed Martin is the major contractor. Lockheed Martin did comply with the DOD’s norms, by following on DOD’s path. Thereby accepting the deviations recommended by the DOD which resulted in delays and failures in the test. This is projected as the systematic building of inertia by curtailing their will to comply with a disciplined approach.

HEO 1 had continuous changes to design due to SPO authorizing to pass the SBIRS High critical design review with just 50% of design drawing whereas the recommended completion was 90%, in addition, IRT report found that program did not invest enough time and resources in basic systems engineering analysis resulting in cost and schedule escalations[Younossi et al., 2008].(***Inadequate control enforcement***). *This is because it gave room for inappropriate creeps due to inadequate enforcement of recommendations*

Major design changes occurred to GEO late in design phase due to technical problems found in testing, in 2000, the HEO flattener lens failed during the first random vibration test and the lens came out of its mounts due to design deficiencies, then the corrector lens failed in the second test in 2001, in addition, due to degraded sensor performance which if left unaddressed would lead to failure to meet KPP, which was resolved by adding 12 foot sunshade for off-axis solar radiation rejection[Younossi et al., 2008].(***Inadequate control enforcement***). *This is because of the failure that was repeated showing the inadequate enforcement.*

As HEO 1 was the first major deliverable for Increment 2, the sensor delivery was delayed by a year from February 2002 to February 2003, further delay was due to first infrared sensor that had significant defects in flight software involving sensor’s ability to maintain earth coverage and track missiles while orbiting the earth in system test in November 2002 resulting in further postponements of delivery[GAO, 2003]. (***Missing control structure***) *This is because of the communication between the sensor development team and flight software team was not structured, resulting in integration issues.*

Further due to signal weaknesses in HEO sensor, which had delayed and increased the risk, SBIRS continued to experience technical issues in GEO signal processing software development and HEO-2 payload software qualification testing after HEO sensor delivery[GAO, 2003]. (***Inadequate control enforcement***).*Requirements of the sensor were not understood properly, so sensor development had many problems.*

As SBIRS High had problems with sensor chip assembly development needed for sensor detector arrays and pointing control assembly software development and control gyro reference assembly also failed during life testing in Increment 1 and ground software problems resulted in two-year slip[GAO, 2003]. (***Inadequate control enforcement***) *This is because of the complexity of the technology.*

There were 148 defective EMI frequencies, which would lead to mission failure, of which 39 design modifications were made and 7 were granted waivers as it would not cause performance problems[GAO, 2008]. There were process failures, stringent requirements and the subcontractor did not implement the EMI control plan which was agreed in EMI design review[GAO, 2008] ***(inadequate control enforcement***). *This is because the improper process was adopted leading to defects.* Further, no contractor was clearly responsible for integrating HEO sensor with the host bus[GAO, 2008]. (***Missing control***). *This is because there was no clear understanding of individual contractor's responsibilities, so DOD did not bother to bring in new processes in place.*

Two late design changes were made to improve GEO satellites success, 80 amp battery was to be replaced with 100 amps battery to improve operational reliability with the estimate of $15 million but the cost performance report in June 2003 shows the contractor was having difficulty assessing the specifications of the battery resulting in schedule delays and increased cost. And the second change was to modify the solar cell panel to resolve power deficiency for which the impact on cost was not yet determined[GAO, 2003].(***Inadequate control structure***). *This is because the design had to change after the clarity of requirements.*

SBIRS High technology integration was a high risk due to insufficient time as by June 2003 only 58 percent of GEO sensor integration, assembly, test and checkout work was completed with $2million of work behind schedule[GAO, 2003]. Software development of ground system efforts was behind schedule by 32 percent by November 2006[GAO, 2007]. (***Inadequate control structure***). *This is because, although there were schedules to complete the sensor integration and ground system development, the schedule did not facilitate the delivery of the products leading to delays.*

The first time integration of flight software was exercised in a new system, the proto-qualification had to be tested using simulators or flight hardware so transition to facilitate the operational use had added complexity in integration and test for GEO 2 as flight software would be operationally used in development testing which led to remaining SBIRS program at risk[Office of the Secretary of Defense, 2005]. (***Inadequate control structure***). *This is because the technology was new so the progress in the process could not be determined.*

Lockheed Martin did project the righteous design completion of 50% but DOD allowed Lockheed Martin to proceed with the development which was the cause of the major design failures and changes. This led to defects in test and design changes due to defects which led to delays and cost overruns.

Designated Approval Authority had not followed the norms by issuing the permission to operate, which had violated **Department of Defense Information Security Certification Authority.** And to allow IHC to operate incorrectly issued Interim Authority to operate, System Security Authorization Agreement is violated.

Defense Information Technology Security Certification and Accreditation

A review was conducted in September 2002 to June 2003 to evaluate development testing of flight software for two ground segments: Highly Elliptical Orbit Intersegment Telemetry, Tracking and Commanding; and Highly Elliptical Orbit Early-On-Orbit Test Mission Processing for completeness, adequacy to testing including planning, execution, and reporting, as a result, validation of security, penetration test of system security features for HEO were found to be incomplete, leading to HEO test data to be doubted and HEO capability to test, assess and support SBIRS was contested[DOD, 2003]. (***Inadequate control enforcement*). *This is because the capability did not undergo the required control structures leading to mission failure.***

The program office did not implement Airforce instruction 99-101 "**Developmental Test and Evaluation**" also SMM tool was not used to track the progress of the program and all critical test plans and reports were not signed off, thus without effective management and oversight of development and testing, the program had the risk of repeating the problems identified during the program recertification[DOD, 2003]. (***Inadequate control enforcement***) *This is because the Airforce instruction did not suit the process of development.*

Systems Engineering Integration Team

GEO P/L configuration is different from HEO P/L and complex, latent defects were discovered in the manufacturing process of HEO in integration and test sequence leading to delays in the schedule[Office of the Secretary of Defense, 2005].(***Inadequate process model***) *This is due to the technology complexity.* System engineering integration team is responsible for validating requirements in test and found integration defects.

Airforce Operational Test and Evaluation (AFOTEC)

The problems uncovered in thermal vacuum testing were not identified earlier, as test beds were developed in parallel had defects which proved difficult to distinguish between testbeds and flight software issues; oversubscription of testbeds and lack of simulation resources had led to high-risk areas such as timing of stored programs which were not tested for insufficient modelling and lack of robustness[GAO, 2008]. (***Missing control structure***). *This is because at every stage there is no check post to evaluate the credibility of progress to the next stage in the process.*

Compressed timeline, issues due to shared facilities at overseas relay ground stations, delay in performance validation testing of increment 1 ground software, inadequate testbed design and scope, simulation tests needed, accelerated deployment of low component in 2004, significant improvement in SBIRS High requirements over DSP and inadequate HWIL testbeds in OT&E has resulted in increased risk and schedule delays and cost overruns[DOD, 2003]. (***Too much of control enforcement***, ***incorrect process model***). *This is because the schedule was very tight and did not incorporate the safety cushion thus pushing problems to the end state.*

Ground software development efforts were delayed due to database problems and the total size of software equivalent lines of code which impacted the schedule in system test[GAO, 2007]. In addition software, development and test efforts had integration and total performance problems due to combined SPA and Pointing Control Assembly hardware and software elements and faulty hardware and software design of HEO/GEO flight computers and problems with ‘halt’ anomalies of single board computer. HEO P/L Single Board Computers had problems, there were three occurrences of unexplained P/L anomaly in halt as all the P/L telemetry data was lost and P/L did not respond to commands in thermal vacuum testing of HEO 1 in 2003[Office of the Secretary of Defense, 2005]. As the telemetry data was lost no conclusions could be drawn and this problem repeated in second P/L thermal vacuum testing as well[Office of the Secretary of Defense, 2005]. (***Asynchronous evolution***) *This is because the delays in the development had affected the test, which delayed further.*

AFOTEC found major defects in the test due to design problems. In effect, the design problems were due to DOD authorizing Lockheed Martin to proceed with development with only 50% design completion. The rushed environment did not have enough resources to carry out the test for the incomplete software.

Department of Defense

The decline in the defense budget, consolidation of the aerospace industry and growing competition for fewer programs, resulting in increased cost efficiency by transferring program responsibilities to contractors with less government oversight[Younossi et al., 2008]. The acquisition reform measures eliminated usual cost and technical risk assessment data recording, increased technological complexity and reduced acquisition workforce due to downsizing challenged the knowledge to assess the technical and system engineering progress of the program[Younossi et al., 2008]. (***Inadequate control structure***). *This is because the efficiency was not indicated as a requirement in the process.*

The DOD has allowed waivers, design completion requirements were bypassed, and test certification procedures were not adopted which had surmounted to building relaxed environment. In addition to these, budget pressure and increased cost efficiency have expected contractors to perform more with less. These had led to inertia in the system.

Constraints categorized for each controller

In *Figure 20* **SPO** had missing control structures in assessing the risks involved in planning schedules, inadequate control enforcement in planning the development of technical specifications and to the extent, the process controls were needed. Inadequate control structures were found in understanding and integrating the progress made in every department; in the risk assessment of the program; all departments were allowed to operate without restrictions, and the delivery of the software were not monitored. Asynchronous evolution was experienced in technical performance which led to schedule slips and cost growth and due to lack of appropriation over the departments.

Airforce had inadequate control enforcement of monitoring the progress of the process. **Lockheed Martin** had inadequate control enforcement on requirements of the project; architectural decisions were not foreseeing the future; requirements were with inadequate clarity; technology complexity had made the project unpredictable; process failures resulted in delays. There were missing control structures due to lack of adequate information about the project beforehand which led to the distribution of contract failures. Inadequate control structures were the result of design changes, schedule delays, the flexibility of the system was planned inadequately. **Defense Information technology security certification and accreditation** had inadequate control enforcement as the resources required were not planned ahead.

Developmental Test and Evaluation had inadequate control enforcement as there was lack of control over the project. **Designated Approval Authority** had not followed correct procedures for the test. **Systems Engineering Integration Team** had inadequate process model in the entire development process. **AFOTEC** had missing control structures to pre-plan for resources. Too much of control was enforced at the beginning of the project due to inadequate planning. Asynchronous evolution was experienced due to cascading effects of delays. **DOD** had inadequate control structures as it could not shield against external influences.

Constraints violations for each controller

Thus in *Figure 20*, **SPO** had inadequate planning, coordination problems, control issues and lack of organizational drive to report the true picture of the status of the program to DOD.

- **Airforce** also had coordination problems and DOD was left uninformed.
- **Lockheed Martin** lacked disciplined approach to the entire program which led to planning issues. These problems were reported to DOD.
- **Defense Information technology security certification and accreditation** could not plan ahead leading to improper resource allocation which was notified to SPO.
- **Developmental Test and Evaluation** lacked control over the project resulting in delays which were notified to Airforce.
- **Designated Approval Authority** had not followed correct procedures and **Systems Engineering Integration Team** had development process problems which were reported to SPO.
- **AFOTEC** had planning issues which were reported to DOD, DOD had reports from SPO, Airforce, Lockheed Martin, and AFOTEC on the program and problems in the process.
- **DOD** had experienced a false sense of the program and visualized the goodness that will shower to advance the program to success.
- **GAO** that is responsible for auditing the program, had reported to DOD and relaxed on advising DOD to take action. So GAO is indirectly responsible for the failure of DOD and DOD is responsible for the failure of the program. So GAO and DOD are considered as missing control structures (represented in dotted lines in *Figure 20*).

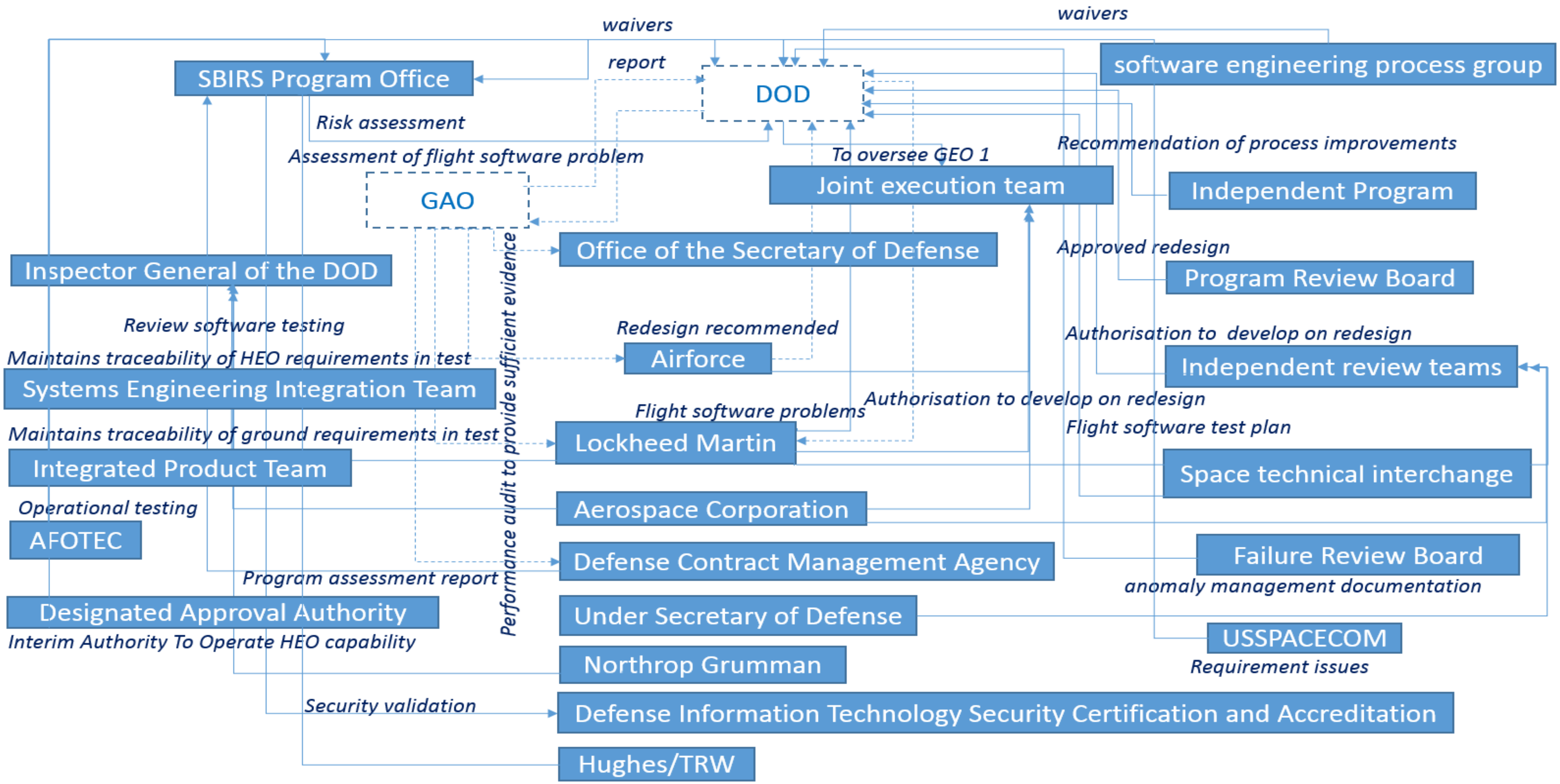

waivers
waivers
report
SBIRS Program Office
DOD
software engineering process group
Risk assessment
Assessment of flight software problem
To oversee GEO 1
Recommendation of process improvements
GAO
Joint execution team
Independent Program
Office of the Secretary of Defense
Inspector General of the DOD
Approved redesign
Program Review Board
Performance audit to provide sufficient evidence
Review software testing
Redesign recommended
Authorisation to develop on redesign
Maintains traceability of HEO requirements in test
Airforce
Independent review teams
Systems Engineering Integration Team
Flight software problems
Authorisation to develop on redesign
Flight software test plan
Maintains traceability of ground requirements in test
Lockheed Martin
Integrated Product Team
Space technical interchange
Operational testing
Aerospace Corporation
AFOTEC
Defense Contract Management Agency
Failure Review Board
Program assessment report
anomaly management documentation
Designated Approval Authority
Under Secretary of Defense
USSPACECOM
Interim Authority To Operate HEO capability
Northrop Grumman
Requirement issues
Security validation
Defense Information Technology Security Certification and Accreditation
Hughes/TRW

Figure 20 Represents the control structured that had disappeared over time and become ineffective. (Dotted lines)

The accident was to place all the applications on a single processor, this accident is the consequence of missing control structures, inadequate control structures, the asynchronous evolution of negligence to deviate from the process over time, inadequate control enforcement, the process model was not understood and the process model in this accident were found inadequate. As shown in *Figure 20* DOD had all the circumstances which led to changes in control structures causing the accident which evolved over a period to the consequence of the accident. It is argued in this research that GAO that has the responsibility of continuously assessing the changes in the control structures is responsible for the accident than the DOD. As it is shown here that the GAO being the auditing wing of government bears the major challenge of taking the program through to success by maintaining smooth operations of DOD. When there are control structure changes that are not adapted to the expected level of rigor required for the continuous process change, it relaxes the expectation of the hierarchical controls leading to negligence which builds over time to result in the accident. GAO over time had set a pace towards the accident by affecting the controls under its guidance, this being DOD. Although there are wider influences, in this case study, only the accident context is considered to analyze the impact of the GAO on DOD which has resulted in the disappearance of DOD in this context.

The missing control structures start with thermal vacuum testing where it is realized that the problems in the flight software were beyond controllable due to the heavy amount of defects that had piled up and an excessive amount of workarounds had weakened the framework leading to a non-recoverable state. The redesign efforts began at that stage which was the consequence of the previous state of the system (i.e. design failure). It will be detailed hereunder, the consequence of the accident was the response of the process failures to continuously evolve towards the accident. The initial stage of the process (the design phase) is proven to be irrelevant to current context thereby the initial context which was right, proves to be eroded over time to the extent that it becomes the root cause of the problem. Here the original design started with two processors with distributed flight software which had been proven as “wrong design” and the new design so adopted was to place flight software on one single processor which was proven as “right design”. In the following sections, it will be shown that the NASA’s ability to scrap the right design and to adopt the primitive design for pacing itself.

Correlation between failures and missing controllers

This section will explain the interaction failures and assign specific controllers to the failed interactions. In the above section, interaction failures were analyzed in every controller and how these failures affected the entire organization and the program was discussed. These failures are analyzed for their influence by connecting the failures on the entire picture of the program that was responsible, will be correlated to the identified missing controllers/organizations. The missing organizations that correspond to the collective failures will be identified in the section below. The inferences (*depicted in italics*) derived from the above section will be detailed to understand the correlation between the collective failures and the identified missing controllers.

SBIRS Program Office

The SPO has performed to its best by proving adherence to DOD's pace, a pace that was set to adopt a relaxed attitude. There were process failures, these failures took the same the path always, creating a pattern of a break in the process, DOD being the end of the process which was left waiting for the process flow to the end. This scenario seems like to have mounted up expectations for DOD, but secretly SPO did know the results of the DOD's initiatives. Unfortunately, SPO maintained a good face of the program. This scenario will be detailed hereunder.

The contractor seems to have taken advantage of the ***inadequate control structure*** in the risk assessment process, as government oversight had reduced, giving the contractors autonomous control on assessing the risk criteria and <u>DOD</u> *was not able to assess the efficiency of individual risk levels assigned to every risk. Thus resulting in the carriage being led by horses at their own whims.*

The program had suffered from the estimation of problems resulting in underestimating the risk and overestimating the efficiency gain in the cost due to ***inadequate control structure*** in *<u>DOD</u> process in the performance of the program efficiency which had led to inaccurate risk levels being graded resulting in failure of process to determine which process has to be prioritized thereby leading to a disastrous outcome.*

The flight software development and integration activities were given high-risk ratings, still, the risks were not mitigated year after year (1996, 1999 and 2004). The lethargy was built in by the time it reached the 2007 thermal vacuum testing failure when DOD was ready to accept any decision to move forward without much thought. *This shows* ***inadequate control structure*** in DOD's *process with inertia built in.*

The schedule variance was very high which led to deferring of certain functions but since flight software could not be deferred until after launch, the pace of software development was increased and more risky functions like integration efforts were postponed. The original problem lies with the assessment of the functionalities required and the schedule baseline to depend on this assessment, which leads to a process of risk assessment that could not assess the risks involved, leading to inappropriate schedule to be baselined. *Thus the* ***missing control structure*** *in the process lies with* DOD's *risk assessment process which did not keep pace*.

The requirement phase was done in parallel with design leading to many changes in design and development did not appreciate the efforts of the design team to cover up for program deficiencies holding the bag for unnecessary complications in the development and test. The program should have adopted rigorous military initiatives. *This is again a clear* ***inadequate control enforcement*** *of processes by* DOD.

Due to technology complexity, the program did not cope with the complexity leading to delays and cost overruns. This is again a case of the ***inadequate process model*** in DOD's *list of process failures.*

SPO had risk assessment failures, risk mitigation failures, schedule delays, requirement clarity failures and design failures. Technology complexity seemed like the big universe which cannot be solved at this juncture. DOD had propagated the lethargy in its constitution which had promoted SPO to stand up in pride to face DOD even with the failures in the program. GAO that is supposed to be auditing and reporting to DOD with recommendations for improvement did attain salvation very early in the program which led to GAO being reluctant of accepting the failures. Thus there was no indulgence from GAO to delve deep into the problems.

Department of Defense (DOD)

DOD being the feeding member of the process in the process loop which was started by itself, the process is the flow to produce the success of the program. DOD does create the loop of processes which flows in and out of various organizations in the different stages of the process. DOD had diligently forwarded the process standards of lethargy which had been propagated through the entire process. This standard was adopted for having a smooth flow of the process, thus the inertia was built in as an intrinsic factor. This will be stated in detail hereunder.

There were process escapes in the program leading to problems being escalated to various degrees resulting in DOD simply turning away from dealing with such situations where a perfect solution is not visible. *This is due to **inadequate control enforcement** of the processes by* DOD.

The delays from program perspective than on individual organizational perspective was an accumulative cause of irresponsiveness which was built in over time. The cumulative effect resulted in mission failure. *This is clearly **inadequate control enforcement** of processes by* DOD.

The program lacked information flow from the contractors to DOD resulting in DOD to be relaxed with the spoon-fed information *encouraging* the DOD *to engage in building a rigmarole of negligence which has led to a state where the program went out of control.*

The compressed schedule led to inadequate software to be delivered to test which led to a cyclic movement from test to development and back, leading to delays and cost overruns. *This is again a clear case of **inadequate control structure*** in DOD's *process to boost the efficiency in the projections of schedule estimates.*

The cascading effect in the process was responsible for the transfer of redundancies from one stage to another leading to inertia in the total system. *This is clearly a case of* a***synchronous evolution*** in DOD's *process.*

Waivers in different stages led to some processes to be bypassed resulting in defects and rework and cost escalation. This is because of the ***inadequate control enforcement** of the process by* DOD.

Two late design changes which did change the pace of the delivery process, which led to an imbalance in the development due to requirement clarification at the later stages. The DOD's process to allow waivers for design and specification to be done in parallel leading to changes in design at later stages. *This is an* ***inadequate control structure*** of DOD's *ability to accommodate new requirements creep which led to compressed process to accommodate change.*

There were many problems in development integration which were not identified beforehand due to unpredicted challenges in technology resulting in complexity gain in various stages of the process. The process itself became responsible for building complexity as intrinsic factor, which had manifested as design problem rather than simplifying the process, DOD had landed up simplifying the architecture. *This is a clear case of* ***inadequate control structure*** *in the process evolution to deliver the program effectively*.

The first time integration of flight software was not well planned leading to complexity in technology being transferred to unplanned processes which led to a hard landing which in turn implanted itself back into flight software development. *This is again a case of* ***inadequate control structure*** in the DOD's *process to accommodate a new process of first-time integration.*

The delays in the development led to delays in the test and the test had to bear the cost of the delays. The design changes brought in by the problems in the test were due to lack of process to transfer the complexity over to next stage in the process that is why there were reverse flows back to design. *This is an* ***asynchronous evolution*** *of the process in the* DOD *towards problems*.

The complexity of technology was seen to cause impact in the development and in turn into testing leading to cascading effect of complexity of technology. *This effect is due to the inability of* DOD's *processes to facilitate the complexity by simplifying the process resulting from* ***asynchronous evolution***.

DOD had managed to pass the inertia as a standard through the process for following processes to take advantage of. This process was adhered to with an ambition to match the pace with each other to form a cohesive conglomerate. This led to a belief that everything is going well and failures were never admitted which did project a sense of oneness in the delivery of the program as a success. GAO could not detect the variances which are where GAO fails to assess the program's progress.

Lockheed Martin

The flight software development team had many integration problems, major integration problems were with sensor development team *which proves that coordination of teams had* ***missing control structure*** in <u>DOD's</u> *processes which led to integration failures and delays.*

The test had to see repeat problems in the design leading to a cyclic response to the process due to ***inadequate control enforcement*** *of processes that should have been adhered to and should have been enforced by* <u>DOD</u>.

Lockheed Martin is the major contractor, had not followed processes that were mandated. These process failures had led to rework and redesign. DOD did not notice the first sign of process failure. GAO was relaxed as it was not keeping a tap on changes in the process flow.

Airforce Operational Test and Evaluation (AFOTEC)

The testing of flight software was not done properly, as the thermal vacuum testing problems were not identified at earlier stages of the test, thereby proving that the control structures in place *could not identify the lack of rigor in the process, thus proving the* ***missing control structure*** in <u>DOD's</u> *testing and development processes.*

Airforce instruction of 99-101 Development test and evaluation was not implemented in the program which was conveniently avoided so that the progress could not be monitored by DOD. This is evident from the yearly voluntary submission of progress report which even DOD had been stunted by such an admirable display of obedience. *This is a clear case of* ***inadequate control enforcement*** *by* <u>DOD</u>.

Some of HEO capabilities were not tested leading to HEO data to be not trusted in the test process which led to confidence loss in the process. <u>DOD</u> *did not enforce the testing certification process properly resulting in* ***inadequate control enforcement****.*

DOD was countered with its own projected standards by very conveniently not following Airforce Instructions in the test which was a stunning response to DOD. GAO seems to be out of the picture.

USSPACECOM

The functionalities that were defined to requirements were not clear enough which led to confusion in the contract management perspective that became more evident when it was realized that no contractor was assigned the responsibility of integrating HEO sensor to the host bus. *There is a **missing control structure*** in <u>DOD's</u> *process which identifies the functionalities from the requirements to contract assignment.*

DOD had all the organizations with assigned responsibilities, but they had not predicted new processes that had to be incorporated which led to a slip in the process that affected the following processes adversely. GAO was not agile enough to detect this slip in the process.

System Engineering Integration Team

There seems to be a clear case of process failure when EMI frequencies were corrected with design modification where the requirements were not clarified, and on top of that, the control plans were not adhered to *which resulted due to **inadequate control enforcement** of processes by* <u>DOD</u>.

DOD's built-in inertia is evident in the SEIT, which is struggling to cope with process failures. GAO was not preemptive to assess the process requirements thereby leading to unplanned loops in process.

Government Accountability Office (GAO)

Due to the compressed timeline, resource limitation, more technology testing, and improved requirements all led to too much of control being enforced in the process leading to escapism in the process which had implanted inertia towards reporting back to DOD. *This led to* <u>DOD</u> to *give up on the program, thus taking off control out of its grips. This is a clear case of the **incorrect process model***.

The processes introduced by DOD had poor performance, but the lapse in the process was not compensated which in effect was carried forward with the program. The assumptions taken at early stages were not corrected to stabilize thereby the complexity was built in as in the case of COTS. GAO *had the responsibility of assessing the process imbalances and still, no action was taken.* (***Inadequate control enforcement***). *This is because of negligence in assessing the requirements.*

Government oversight was not managed appropriately leading to contractors taking advantage to produce a biased estimation of cost and schedule. Later to meet the cost and schedule, processes were granted waivers which increased program risk. GAO *did not stop this evolution of inertia in the process.* (***Asynchronous evolution***). *This is because of negligence to adhere to the set process leading to problems in assessment processes.*

GAO has had a tremendous lapse in assessing at various stages leading to inappropriate standards being set for assessing the information required to be assessed being moderated according to the situation, there seems to be more brewing than what was actually reported for assessment. The real sense of negligence has been portrayed by GAO for others to make use of the situation for their benefit. GAO has encouraged such a false self-esteem to be their honor. In favor of GAO, which has the name to be righteous and demanding that has promoted other organizations to be over submissive taking the strides to the limits of other organization's capabilities which has led to the suppressed deceptive outpour. DOD was caught is a line of fire of GAO and unfortunately, DOD had lost its perseverance to adapt to changing expectations. DOD seems to have been a very good partner in crime.

It is the GAO that had been indirectly responsible for the failure of DOD leading to the accident. GAO gets DOD to disappear from the program while itself merging in and losing its projected pride. Standing by GAO, the land does seem far offshore than one could dream of, it is an understanding that GAO did try to cope with the other's expectations to please everyone. It is time for the volcano to erupt, which will bring severe GAO to light by sanctioning honor to be projected, hopefully in due course.

Thus *Figure 20* projects the controllers that had failures which resulted in an accident. These failures are mapped to controllers as inferred from the above section in the table below (*Table 1*).

SPO problems →Risk assessment problem →Prioritization of mitigation efforts →Improper handling of escalated risks →Improper schedule baseline based on risk assessment →Inadequate process implementation **Lockheed Martin** →Inadequate process to handle integration problems →Inadequate reverse process to solve the identification of problems **Airforce Operational Test and Evaluation (AFOTEC)** →Inadequate process in performance monitoring of the process →Inadequate and timely corrective action to bring confidence in the process **USSPACECOM** →Inadequate process of assigning contractors to functionalities **System Engineering Integration Team** →Inadequate process monitoring	**DOD** problems →Turning away from dealing with problems →Inadequate initiatives to motivate the organizations to work together →Assumed goal to reach the efficiency →Blame culture of transferring problems to some other section →Improper enforcement of processes →Inadequate flexibility of processes →Identification of solutions to smooth running of the program →Inadequate flexibility to adapt to the change →Inadequate planning of defect handling process →Inadequate process to handle technology complexity →Cumulative failure resulted in accident **GAO** →Too much of control on the process →Inadequate assessment of process failures →Inadequate monitoring of the processes

Table 1 Interaction failures mapped to controllers

SPO is responsible for the entire program’s success, which had problems with the risk assessment that was dependent on various other factors as discussed earlier. As risk assessment was the area where failures were noticed, the program had the cascading effect on design, development, and test. Lockheed Martin being the major contractor was responsible for development and test of the software, had problems with coordinating with other teams and the effect of risk assessment led to problems in development life cycle leading to cyclic movement of the blame. AFOTEC had process problems in the operation of the test. The test had the cascading effects of risk assessment from the development and design. USSPACECOM had missed requirements to map to the contractors which were realized later in the process. Thus the process was not monitored to ensure appropriate procedures were followed. SEIT had the same problem as in USSPACECOM. DOD had process failures in dealing with technology complexity, defect handling, adopting change processes, smooth execution of processes, flexibility to adopt new processes, process adherence problems, coordination processes, realistic planning process, the process to mediate goal between organizations and not enough resources to deal with problems. These process problems had accumulated to form an accident. GAO’s restricted constraints led to an artificial show of obedience which led to failures in the monitoring of process which in turn led to assessment failures in auditing.

Thus GAO initiated the environment and DOD adhered to the GAO’s expectations. The organizations responsible for the accident are DOD and GAO. So the in *Figure 20* DOD and GAO are represented with disappearing controllers as their influence or noninfluence had caused the accident.

The actual cause of the accident is still not clear at this stage of STAMP analysis as the controllers are analyzed by using static snapshots. Although the analysis points to the organizations responsible for the failures, their influence on the accident is not clear in the overall picture. The clarity of the cause of the accident is derived from the dynamic process which will collate all the controllers and failures and link it to the accident. In the next section, the link between various failures leading to the accident will be analyzed.

3.4.3 Dynamic Process Model

Dynamic process model will prove the importance of GAO's role in auditing and monitoring the process. GAO had the constraints and controls in place to monitor the system, but over time these had degraded. The system had evolved towards the state of the high risk that even the slight change would result in a catastrophic failure. The changes in the system are complex and the resulting system dynamics are unpredictable. To have an understanding of accident prevention, an analysis of static structures alone will not suffice, the dynamic influence of the system which is an intrinsic factor has to be understood. The system will describe the dynamic nature of changes that are brought and the responsive changes that the system is under. The theory behind this change of the system which is underlying causes the influence on the system which has to be understood for prevention of negative effects on the system as every system has its own context which influences the effect of the change.

Figure 28 shows the system dynamic model for the SBIRS accident. The basic structures in the model are variables, stocks (represented by small rectangles), and flows (double arrows into and out of stocks) and phases in development life cycle (represented by large rectangles). Lines with arrows between the structures represent causality links, with a negative polarity means that a change in the original variable leads to change in the opposite direction of the target variable.

The system which is influenced by the feedback loops over time degrades and they are balanced by the government regulations and oversight which controls the behavior of the influenced system. The system here is influenced by other organization's expectations. They are under the "pleasing factor" influence, which is when a new initiative is introduced, every organization involved is equally motivated to see the success banner of the initiative, so the initiative is not evaluated for its merits rather for the mere satisfaction of creating a success outlook. The system which is influenced to please others is more fragile as the true outlook of the entire system is very difficult to evaluate.

GAO had the responsibility of assessing the progress of the program, GAO had analyzed the progress and reported that contractor was showing overly optimistic progress estimation as troubles in assembly, integration, and test resulting in overrunning cost and schedule was evident[GAO, 2007]. As ground software development was accelerated, databases delivered late had affected the development, GAO had assessed and reported the capability of the contractor to deliver databases[GAO, 2007]. GAO had expressed concerns on DOD's workforce reduction, which had affected the cost analysis[GAO, 2007]. GAO had warned DOD of previous satellite programs that had taken longer and had cost higher due to inadequate performance requirements defined at the beginning but there were many changes in performance requirements which led to schedule overruns in SBIRS program as well[GAO, 2007]. GAO had strong recommendations to DOD of achieving stable design before entering product demonstration but DOD did pass the critical design review with only 50% of design completion resulting in major changes to design at later stages[GAO, 2003].

The SBIRS program had been restructured several times due to the cost increase, schedule delays and revised goals in 2002, 2004, and 2005[GAO, 2008]. GAO had documented all the problems in the processes but DOD had turned blind to its warnings[GAO, 2007].

The failures in the organizations were extracted out to understand the relativity to the development lifecycle in order to analyze the dynamic nature of failures affecting the system leading to the accident. The list below explains the failures related to the individual phases in the development lifecycle. These failures in the development cycle are depicted in *Figure 28* to understand the dynamic process model of STAMP accident analysis.

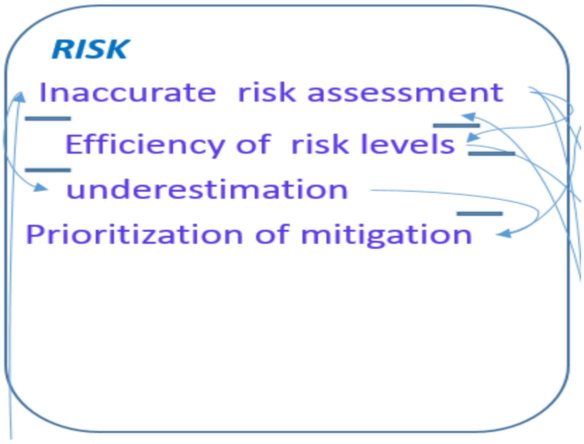

RISK
Inaccurate risk assessment
Efficiency of risk levels
underestimation
Prioritization of mitigation

Figure 21 Risk-Dynamic process model

Risk (**Figure 1**Figure 21)

- As government oversight had reduced, giving the contractors autonomous control on assessing the risk criteria leading to *inadequate risk assessment process*.
- DOD was not able to assess the efficiency of individual risk levels assigned to every risk leading to *inappropriate risk categories* defined.
- *Underestimating the risk* and overestimating the efficiency gain in the cost.
- The flight software development and integration activities were given high-risk ratings, still, *the risks were not mitigated* year after year (1996, 1999 and 2004).
- The pace of software development was increased and more risky functions like integration efforts were postponed resulting in *schedule variance*.
- Risk assessment process that could not assess the risks involved leading to *inappropriate schedule to be baselined*.

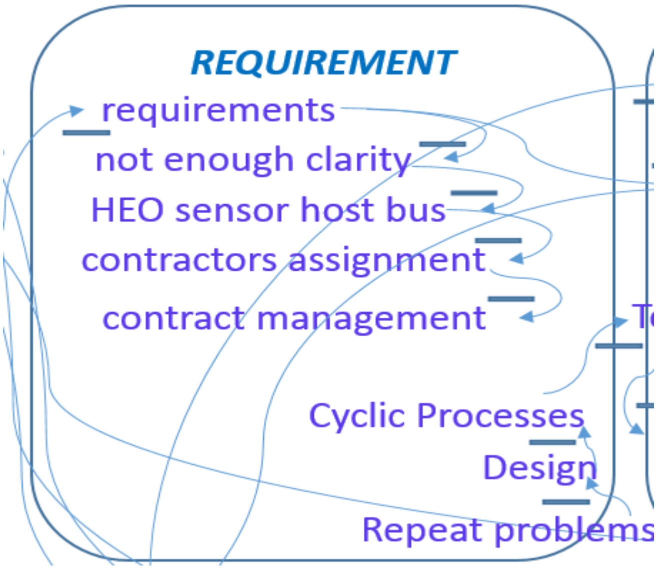


Figure 22 Requirement – Dynamic Process model

Requirement (**Figure 22**)

- The requirement phase was done in parallel with design leading to *many changes in design and development.*

- The test had to see *repeat problems* in the design leading to a *cyclic response to process.*

- The program did not cope with the complexity leading to *delays and cost overruns due to technology complexity*.

- There were *process escapes* in the program leading to problems being escalated to various degrees.

- The delays from the program resulted in *irresponsiveness* which was built in over time.

- The program *lacked information flow* from the contractors to DOD resulting in DOD to be relaxed with the spoon fed information.

- The compressed schedule led to *inadequate software* to be delivered to test which led to *a cyclic movement from test to development* and back, leading to delays and cost overruns.

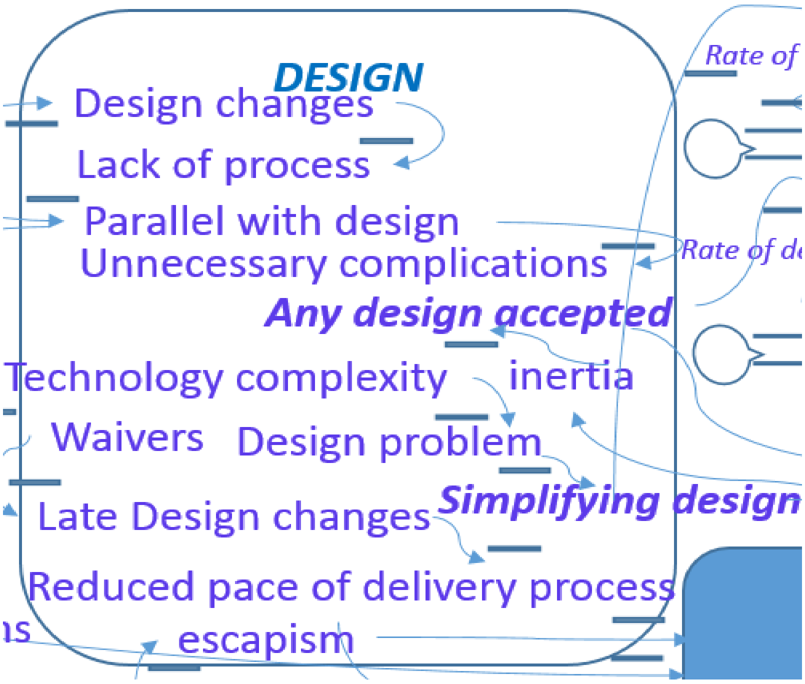


Figure 23 Design – Dynamic Process Model

***Design** (Figure 23)*

- *Waivers* in different stages led to some processes to be bypassed resulting in defects and rework and cost escalation.

- *Two late design changes* which did change the pace of the delivery process, which led to an imbalance in the development due to requirement clarification at the later stages.

- The DOD's process to allow waivers for *design and specification* to be done in *parallel* leading to *changes in design* at later stages.

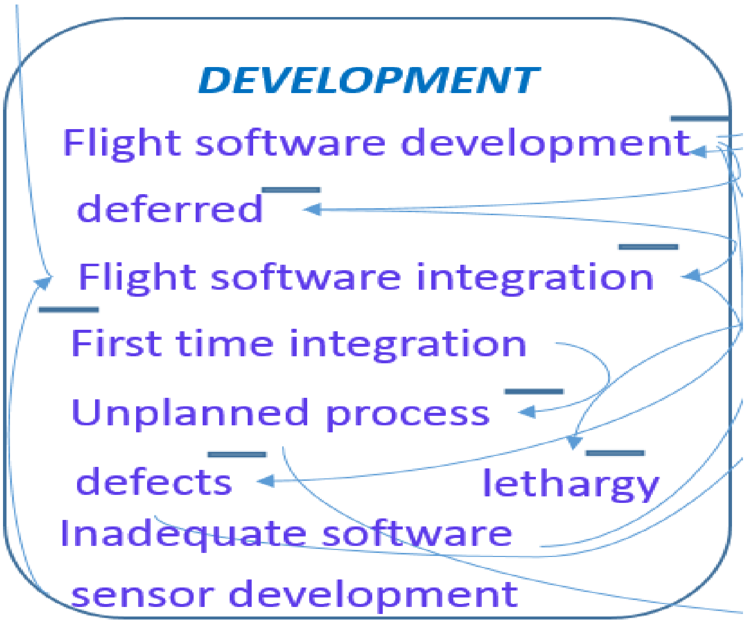


Figure 24 Development – Dynamic Process Model

***Development** (Figure 24)*

- There were many problems in development integration which were not identified beforehand due to *unpredicted challenges in technology* resulting in *complexity* gain in various stages of the process.

- The flight software development team had many *integration problems*, major integration problems were with *sensor development team.*

- The *first time integration* of flight software was not well planned leading to *complexity in technology* being transferred to *unplanned processes* which led to a hard landing which in turn implanted itself back into flight software development.

- The functionalities that were defined to *requirements* were *not clear enough* which led to confusion in the *contract management* perspective that became more evident when it was realized that no contractor was assigned the responsibility of integrating *HEO sensor to the host bus.*

- There seems to be a clear case of *process failure* when EMI frequencies were corrected with *design modification* where the requirements were not clarified, and on top of that, the control plans were not adhered to.

- The process itself became responsible for building complexity as intrinsic factor, which had manifested as *design problems* rather than simplifying the process, DOD had landed up simplifying the architecture.

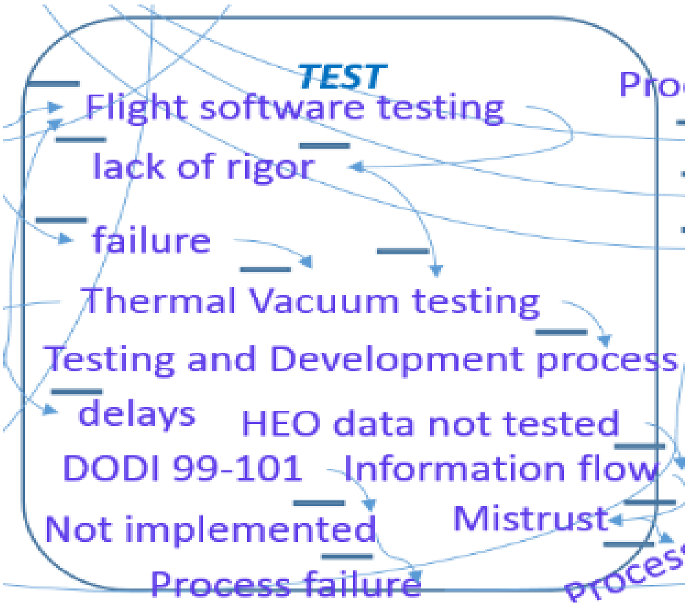


Figure 25 Test – Dynamic Process model

***Test** (Figure 25)*

- The delays in development led to delays in test and test had to bear the cost of the *delays.*

- The testing of flight software *was not done properly,* as the thermal vacuum testing problems were not identified at earlier stages of test

- *Airforce instruction of 99-101 Development test and evaluation was not implemented* in the program which was conveniently avoided so that the progress could not be monitored by DOD.

- Some of *HEO capabilities were not tested* leading to HEO data to be not trusted in the test process which led to *confidence loss* in the process.

- The design changes brought in by the problems in the test were due to lack of process to transfer the complexity over to the next stage in the process that is why there were *reverse flows back to design*.

- *The complexity of technology* was seen to cause impact in the development and in turn into testing leading to cascading effect of technology complexity.

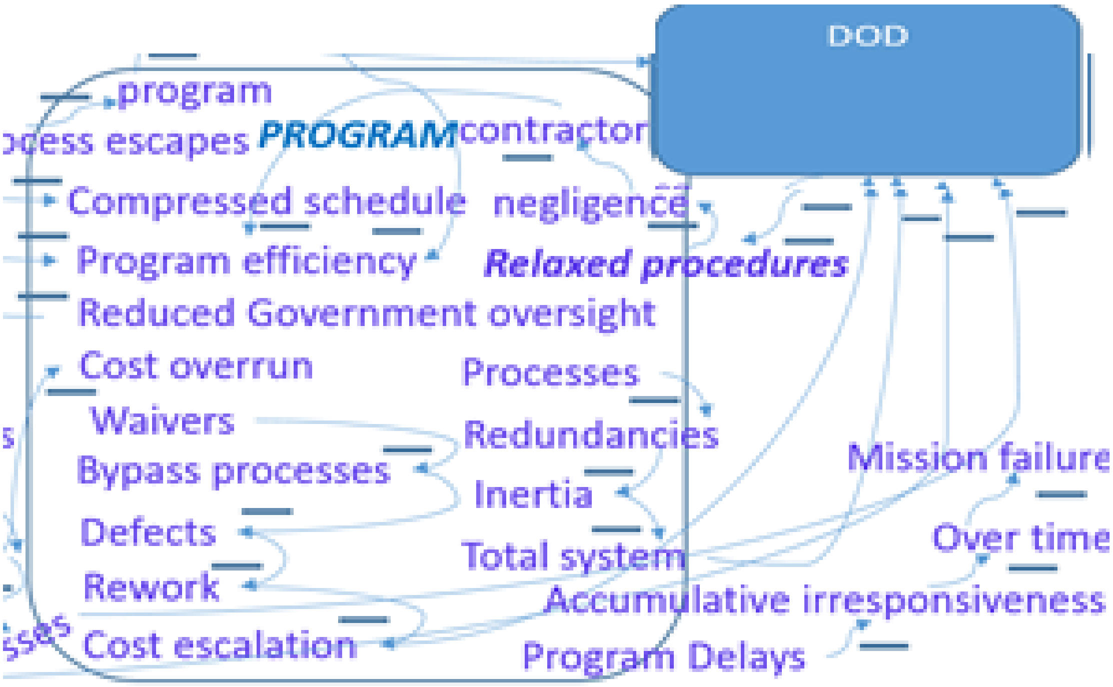


Figure 26 Program – Dynamic Process Model

***Program** (Figure 26)*

- Due the compressed timeline, resource limitation, more technology testing, and improved requirements all led to too much of control being enforced in the process leading to *escapism in the process* which had implanted *inertia* towards reporting back to DOD

- The processes introduced by DOD had *poor performance*, but the *lapse in the process* was not compensated which in effect was carried forward with the program.

- The assumptions taken at early stages were not corrected to stabilize thereby the *complexity was built in* as in the case of COTS.

- Government oversight was not managed appropriately leading to contractors taking advantage to produce a *biased estimation of the cost and schedule.*

- To meet the cost and schedule, processes were granted *waivers* which *increased program risk.*

- The *schedule variance* was very high which resulted in the increase in the pace of software development and more risky functions like integration efforts were postponed. The original problem lies with the assessment of the functionalities required and the schedule baseline to depend on this assessment leading to *inappropriate schedule to be baselined* which resulted in *cost overrun.*

- There were *process escapes* in the program leading to problems being escalated to various degrees resulting in DOD simply turning away.

- The cascading effect in the process was responsible for the *transfer of redundancies* from one stage to another leading to *inertia* in the total system.

- Waivers in different stages led to some processes to be bypassed resulting in *defects* and *rework and cost escalation.*

- There were many problems in development integration which were not identified beforehand due to *unpredicted challenges in technology* resulting in *complexity* gain in various stages of the process. The process itself became responsible for building complexity as intrinsic factor.

- The test had to see *repeat problems* in the design leading to a *cyclic response to a process* which led to defects which were transferred to design phase and back to the test.

- The functionalities that were defined to requirements were not clear enough which led to confusion in the contract management perspective that became more evident when it was realized that no contractor was assigned the responsibility of integrating HEO sensor to the host bus which led to *program inefficiency*.

- The processes introduced by DOD had *poor performance*, but the *lapse in the process* was not compensated which in effect was carried forward with the program resulting in *negligence* being built in the process.

As shown in *Figure 28* Risk, Requirement, Design, Development, Test, and Program are phases of development life cycle. SBIRS is DOD's program initiative. Architectural failures and Redesign of software architecture are the stocks. Although there were many failures accumulated towards the accident, one such failure link represented in *Figure 28* is explained. Inaccurate risk assessment leading to inefficiency in determining the risk levels, which led to an underestimation of risks thereby leading to problems in the prioritization of mitigation in the **Risk phase** of the life cycle.

This prioritization of mitigation led to requirements problems which were due to the requirements with not enough clarity, as HEO sensor host bus was not assigned to any contractor which was due to contractor management problem in the **Requirement phase** of the life cycle.

These requirement issues led to unnecessary complications as requirements were done in parallel with design in the **Design phase**. These complications in the design phase led to problems in the flight software development, which led to flight software integration problems and resulted in defects in the **Development phase** of the life cycle.

These defects led to design changes which were not planned (lack of process). Flight software development defects had failures in thermal vacuum testing. The repeat problems in **Test phase** led to lethargy as flight software development which underwent design changes as the problems repeated, which led to cyclic processes thereby leading to technology complexity due to design problems. Later when there was no other path ahead, the simplified design was opted for. Flight software underwent a redesign of software architecture. The rate of redesign failures had led to architectural failure as the redesign was based on single core architecture which would in future lead to accidents (*Figure 27*).

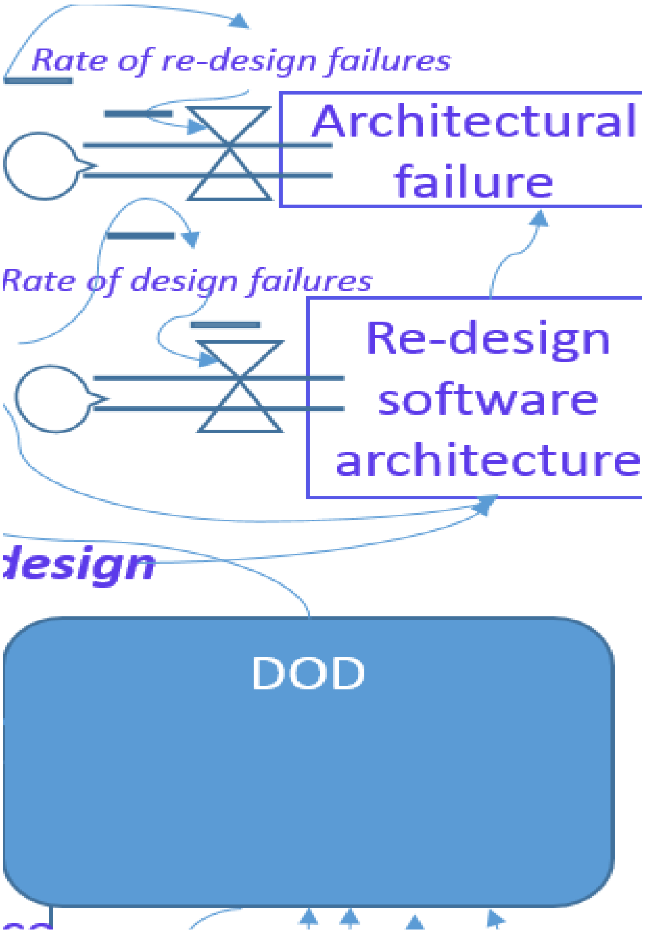


Figure 27 Accident – Dynamic Process Model

The **Program phase** of the life cycle had problems due to DOD initiating relaxed procedures which encouraged negligence in managing the contractor that led to reduced program efficiency. The reduced efficiency led to program delays which in turn led to accumulative irresponsiveness over time. This will result in mission failure.

Thus it was identified that the system "Context" was the cause of the accident. As all the development phases of the life cycle had failures, the only common factor found, that did not attain equilibrium with controllers was the system context. The system context was the only unchanged factor in the analysis which remained constant in the case study.

This mission though did not fail. The current flight software architecture is based on multicore processor and has distributed application architecture. Thus the flight software architecture must have undergone couple more iterations of the redesign of software architecture to attain this refinement in the architecture. And there must have been couple more accidents before this refinement in architecture was attained. These accidents are never noticed, as the end result of the program is successful. This accident analysis is to understand those failures that are not identified as accidents and learn from these accidents to improve the efficiency of the program and to efficiently allocate resources to reduce cost overruns.

Figure 28 shows the relationship between failures which led to the accident. In order to depict the dynamic relationship, the failures are extracted out into the development process to show the dynamism of the failures to result in an accident. Here the interaction context is understood based on the STAMP. While understanding the interaction context, the context of the system as a whole is validated against the interaction context. The system context seems to have a strong bearing on the controllers, constraints and the interaction context. As the process originated from the DOD, the underlying context of the program bore its signs of conformity towards the passed on valor. This became the intrinsic standard which every organization was sublimed to accept and follow. This is when the analysis was encouraged to widen the scope of understanding of the system context. When the system contextual factors were analyzed, a realization of coaching the system context would help to improve individual morale was understood.

It is the realization of the organizational context that was derived from the accident analysis which encouraged the use of contextual factor as the prevention technique. The context is an incubator of deriving self-actualization and stabilizing the equilibrium between the past experiences and to see beyond the future. To seduce this context though is very difficult, it has to be coaxed by the enthusiasm to attain satisfaction of oneself to press ahead of the future. The aim is to provide the direction of the individual satisfaction to derive the excellence in favor of their own sense of esteem. The organization is made of the consciousness which provides a sense of victory in standing together in oneness. This victory is what to be achieved to prevent accidents from happening in future.

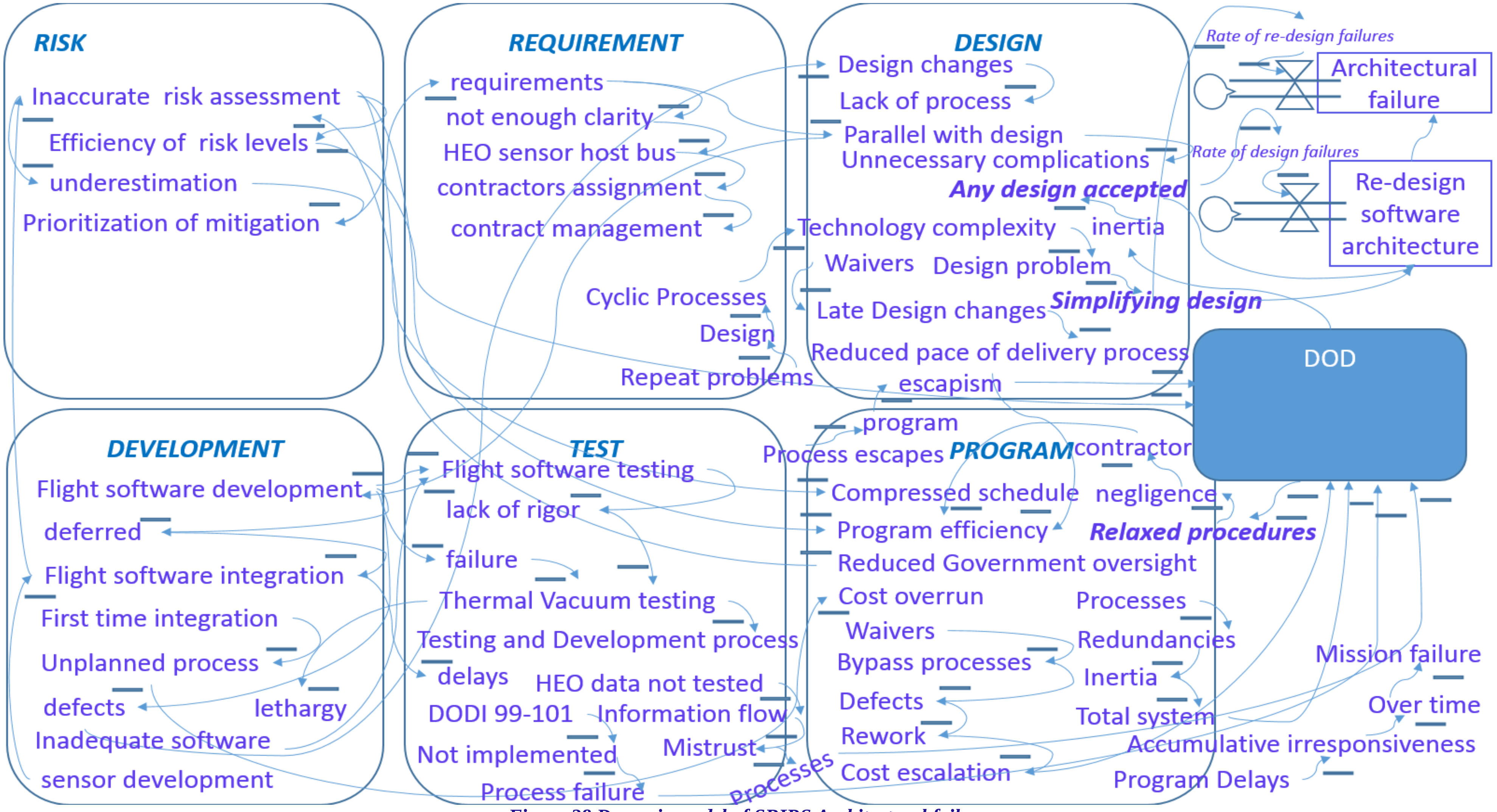


Figure 28 Dynamic model of SBIRS Architectural failure

An accident occurs when the underlying context is shaken (*Figure 29*). The organizations in the context are automatically adjusted to fit themselves as self-adjusting measure - a coping mechanism. In reality in such a well-coordinated system, it is difficult to isolate the good functioning system out, therefore difficult to identify the origin of the accident or even have the realization that system is heading towards failure. In this case study, GAO monitors the context of the system and the accident is the decision taken to alter the architecture in the wrong direction, which is right to the current contextual status. So none of the organizations in the context ever had the realization that the accident had taken place.

In *Figure 29* system contextual factor is added to STAMP core principles (*Figure 15*) as the context has influenced organizational control structures and constraints that maneuvers the behavior of the process to form an equilibrium between the changing constraints and the interaction between the hierarchical levels of control. As the change is a continuous process the process flow which interacts with the constraints to form an equilibrium, the progress of process flow influences the control structures and thereby the resultant of the flow. This resultant depends on the context to lead the project to success or failure. This context is what to be understood and monitored. This context is added in *Figure 29* to STAMP core principles to analyze the accident prevention techniques.

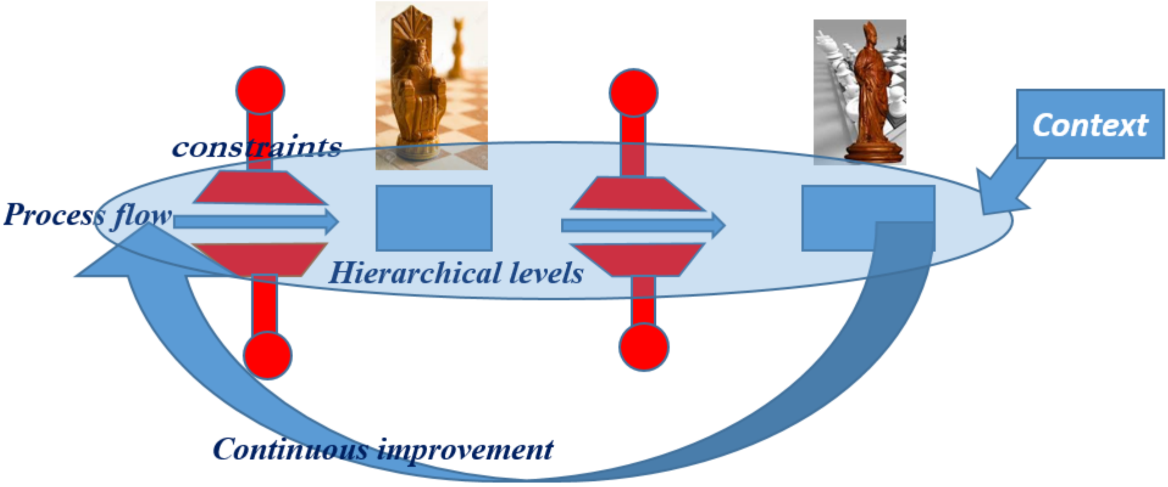


Figure 29 Organisational Context

The analysis carried out here will state the "*state of context*" before the accident, which led to the accident. To prevent such accidents from happening, it is important to understand the contextual state so that changes in the controls and constraints could be predicted.

In the case study, inaccurate risk assessment led to prioritization of mitigation issues which started with inaccurate requirements with not enough clarity which led to contractor assignment issues, this issue was realized only in the integration, when there were excessive amount of defects which were pushed back as design problems, fed-up with the repeat problems from development, test team pushed it back to design, there was a cyclic process established. This cyclic process created frustration among all the teams involved, leading to technology complexity as problems could not be resolved in development. This back and forth in the process had developed unwarranted processes such as workarounds, skipping some test procedures, this led to process redundancies which resulted in irresponsiveness overtime.

At this state of the system, the system was ready to accept any changes which will set the sailing course, so the old design was scrapped and a new design was adopted which is the architectural failure this case study has investigated. In actual course of evolution, waivers were blamed to be the cause of this tornado, as it swirled through design and requirements as parallel processes, development test and testing simultaneously resulting in the cyclic process of repeat problems.

DOD brought new initiatives, oversight was reduced (effect of TSPR), procedures were relaxed and waivers were granted in development processes which led to bypassing certain processes, resulting in rework and cost escalations thereby reducing the pace of delivery and finally the program was on the brink of failure. This is when the simplified architecture was suggested, which came as an escape mechanism from the situation. DOD and GAO authorized the design. In the next section, complete description of why this design was wrong will be detailed and the reason why it is considered as an architectural failure will be analyzed.

It is understood that the context of the accident is the most prominent factor to be analyzed in an accident. As the constraints and controls were changed in the context and response could be seen as a flow of the process, while this process is a continuous process which brings continuous improvement, the only factor that needs to be controlled would be the context.

The Columbia investigation report identified “broken safety culture” as the cause of the accident. The structural secrecy is built into the organization leading to failures[Qureshi, 2008].

Schein refers to *"the culture of a group can now be defined as a pattern of shared basic assumptions that the group learned as it solved its problems of external adaptation and internal integration, that has worked well enough to be considered valid, and therefore, to be taught to the new members as the correct way to perceive, think and feel in relation to those problems."*[Qureshi, 2008]

3.4.4 "Context" As Cause of Accident in other case studies

"I believe that this nation should commit itself to achieving the goal, before this decade is out, of landing the man on the moon and returning safely to earth" John F Kennedy in an address to Congress 25 May 1961.

The report on Columbia space shuttle accident was reported in "*The Nimrod Review*" as organizational context problem. There were lessons learned from various other accidents based on "context" as the cause of the accident. The accidents from RAF Nimrod aircraft, Piper Alpha, Kings Cross Fire accident, Space Shuttle Challenger and Columbia and BP Texas City [Spence, 2009].

The Columbia Accident Investigation Board realized that fundamental element of the success of any project is "organizational causes", so rather than just focusing merely on errors and omission by individuals, the context should gain focus in any project to be successful[Spence, 2009].

There is certainly a message emanating from the accidents in this era, organizational features are the most consistent factor in shaping the success of any project. Technology has helped to improve various aspects of the organizational agility. Agility is balancing the historical perspective and future perspective to develop a defense mechanism against deterrent factors that drive the organization towards the ineffectiveness.

A subservient organizational context is dangerous as it demands compliance to the factors that influence which forgoes the individual strengths and conforms to the expected norms resulting in the complacency of the influencing factor which over time results in irresponsiveness to the instinctive behavior which leads to false sense of valor. This pattern is repeated bringing much more under its influence.

An organization needs a sampler of organizational context who would be passionate about the opportunities of the organizational strengths and would predict the future of trends set by innovations for future to prove the success of more innovations to follow the trend. This sampler is no more a top official, he is the one who feels the pulse of strengths and aims to strengthen the strengths of the organization innovatively.

3.4.5 Summary of Accident analysis

The accident cause is the context in which the redesign was introduced. DOD had taken steps to confirm with program review board and independent review team before approving the redesign efforts. DOD had undertaken all measures for the smooth execution of the program. GAO that has the responsibility of assessing and recommending the redesign options had analyzed the context of the system and chosen this recommendation of redesign as the solution. From this STAMP analysis, it was determined that no specific organization is responsible for the cause of the accident but the context in which all the organizations operate are treated as a causal factor.

DOD had many process failures, TSPR was brought in with efficiency as the background gain by cutting the bureaucratic procedures. But it turned out to be adverse, with less oversight on contractors, and in SAMP, documentation was reduced which missed the details needed. The risk assessment strategy of CRIMS was also a failure as it did not consider proper risk assessment criteria’s in COEA, the cost-effectiveness was not based on the performance, so the cost was not estimated on the real risk. The joint execution team was supposed to oversee the progress of the program, which was also a failure as Airforce did not involve in the progress of the program, so the rein was left to the contractors. The requirements were not properly clarified as USSPACECOM had compromised on performance tradeoff. Contractor Logistics support which was assigned to contractors did not work on the Airforce terms. SEIT was not properly equipped as the TEMP could not predict the risks undertaken by the development team. System Evaluation and Estimation of Resources did not carry out software technology risk assessment properly, so there were problems in development.

There were operational failures in the process, the DOD had granted waivers for many crucial processes like design and requirements were done in parallel, development testing, and testing was done in parallel and design was passed into development with just 50% completion which resulted in many major design changes.

There were major defects in the test, which were counteracted with workarounds which later mounted to be major failures in the test that only design changes could stabilize the software. Some of the certification procedures were bypassed which led to the inadequate testing of software. SPO had limited ability to oversee the progress of the development due to lack of an integrated system to assess the schedule and cost overruns. COTS were assumed to be a low risk which was the cause of major problems in HEO.

All the process failures were complicating the system's analysis of the real problem. The problem of redesigning the software architecture for the sake simplifying the process was the accident analyzed here in the case study. The redesign was wrong to place the flight software on the single processor and DOD did still approve the redesign.

DOD had taken all measures to have smooth execution of the program, but it does seem that the processes had failures which were either not reported or not taken any action on, thus it led to inertia built into the system.

GAO had reviewed technical documents on flight software and conducted performance audits at the office of secretary of defense, space, and missile systems center, Lockheed Martin, Defense contract management agency and had assessed various alternatives for mitigating flight software problems found in thermal vacuum testing in 2007 and had developed a way forward to implement redesign efforts. DOD had approval from program review board for a new design for revised cost and schedule. And six independent review teams examined the new design and authorized to proceed with formal coding.

GAO had analyzed the context of DOD, Lockheed Martin and DCMA which was in a state of complexity that no way forward could be seen. So GAO had recommended the new design to DOD which suited the context well. And GAO was proven right over time as JPL developed Mission Data Systems architecture and GSFC developed core Flight software which holds the pride of the entire software industry in bringing flight software architecture to the dais of architectural achievements.

STAMP analysis was chosen as it is based on system theory which considers accidents as arising from the interactions among system components and usually does not have a single causal factor.[Leveson et al., 2003] As STAMP considers inadequate control or enforcement of constraints on the design, development and operation of the systems, accidents such as these, involving software architectural design errors, may stem from inadequate control over the development process, i.e., if risk is not adequately managed in the design and implementation processes[Leveson et al., 2003]. So the STAMP was chosen for the accident analysis of this case study. The role of the control in the form of external factors such as political factors should also be considered in accident analysis[Leveson et al., 2003]. Thus accidents are viewed as flawed processes involving interaction among system components[Leveson et al., 2003]. STAMP is considered from three basic concepts: Constraints, hierarchical levels of control and process models.[Leveson et al., 2003]

In the case study, the processes failed, control structures were changed and constraints were modified. The STAMP accident model describes these factors as causal factors. The context in which organizations operate is what is found as a causal factor in this case study. DOD did implement all possible measures for the smooth execution of SBIRS program, but still, the processes failed and control structures had changed and constraints such as military instructions were not followed. These factors could have led to accident independently, but collectively they had a major impact- a loss of $7.8 billion. GAO that was to assess the progress could not detect major problems earlier. So the context is blamed for creating irresponsiveness resulting in building lethargy and thus inertia.

The data gathered from the GAO, DOD, and SBIRS program office, Office of Undersecretary of Defense, Airforce, Lockheed Martin, and NASA, were analyzed for influences of the context of the accident (*Table 2*). The contextual factors were examined and the relevant factors were weeded out to relate the accident to the context. The accident context was derived from the context to ascertain the accident causation. As the context is accident cause, this context was separately analyzed and accident context was understood from it. This exercise was undertaken as the accident was a very primitive problem (architecture), with an assurance that NASA would never have considered this architectural decision to be adopted, which encouraged this research to understand the entire program context to analyze the circumstances in which such adverse decision was undertaken. This research has analyzed 15 years of data to derive this accident analysis.

As it is DOD’s context, DOD could have saved this failure by modifying the context as shown in *Figure 30*. It is assumed that if a context is supplied, the other factors might hold the safety boundaries. So if the first process in the organization is standardized with the required context, the same could be replicated in the following organizations.

Figure 30 shows organizational context for organisation1 and organization2. The process which flows through one organization to another should contain the successful context of the project. This context should be enforced into new organizations to ensure the program’s success is replicated in all the organizations that are involved in the program.

The context is recommended to be of military valor which promotes the feeling of pride in the execution of the project in the process. This is every person’s self-esteem which finds the footing in the progress of the project which sees their potential to attain self-actualization as the project progresses and succeeds. The individuals in the context are to be seduced with project focus to derive individual satisfaction to attain self-esteem. This context has to be deployed in one organization and standardized successful context has to be passed on to other organizations.

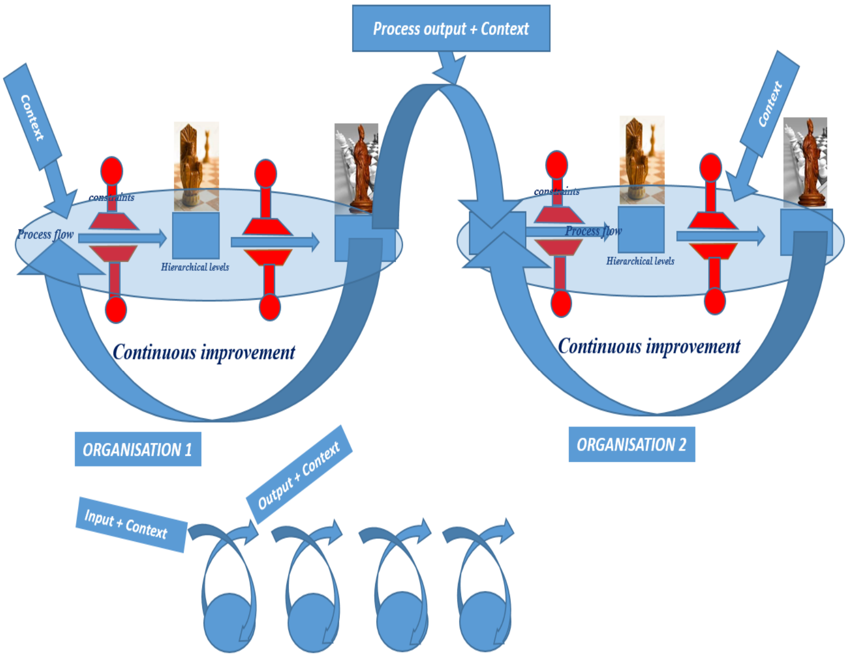


Figure 30 Recommended organizational context enhancement

For example, DOD should have passed context of pride, excellence, potential to excel to higher ranks, created an environment of technological supremacy, honored the role models and created an environment of courage and conviction in general context. The project-specific context would be the drive towards the project excellence, knowledge gain, clear goals, clear roles and responsibilities and accountabilities. This coaxing should motivate individuals to find their goals to attain satisfaction which would be the first step.

	Context Factors, Accident Context, Program Problems, Architectural Problems		
	Problems identified in reports	*Problems identified by this research*	*How these factors contribute to accident*
Program Level Problems	DOD directives 5000 series were not followed	promising system concepts	architecture so adopted did prove to be a drastic failure
	Rigorous to set initial baseline for system performance requirements and KPP	Basic requirements of the mission were foregone.	to meet the schedule and cost
	SAMP was the document which collated all aspects of the acquisition	was not appropriate for the military context	as more stringent measures were foregone and expressive nature of the various aspects of the program was eliminated
	Phase I Pre-EMD did not define the design concepts to the defined requirement	many changes in the engineering design as the program progressed	ineffective at that stage of adoption
	requirements were not properly defined	resulted in formulating a high-level architecture	overthrown for its own good
	COEA was not effectively performed	cost-effectiveness was derived from the performance analysis	Predictability of software delivery
	performance analysis which was not performed on the grounds of achieving higher efficiency	rather it was based on amicable solution among the Defence organizations	Software quality problems
	Cost as Independent Variable was developed to maximize the military utility for affordable KPP	many changes to the specifications in battery and power generation aspect of the design	but the affordability ran out of the logical premise
	The greatest success of SBIRS characteristics was contractor empowerment[Jay A. Moody, 1997][Jay A. Moody, 1997][Jay A. Moody, 1997][Jay A. Moody, 1997], enhanced communication, reduction in overhead, relaxed documentation and reduced government oversight	- contractor empowerment led to the improper assessment of technical efficiencies which led to -cost estimation errors, - reduction in overhead led to improper cost estimation as the personnel were not trained in the risk factor analysis, - reduced government oversight led to improper coordination between the ground and flight team	together led to unplanned changes in design and many estimation errors in specific technical errors
	Contractor Logistics support was considered to eliminate military personnel from maintaining SBIRS ground infrastructure	resulted in expensive coordination problems	proved as a wrong decision from the government part to hand over the infrastructure to contractors
Requirement Problems	Clarity of the requirements was in question at every stage	Re-requirement analysis for understanding the operational clarity.	The workarounds mounted up to a stage where it started its Pareto-optimal tendencies resulting in redesign effort in 2007

Architecture problems	CRIMS was developed for technical risk assessment process	technical risks were not assessed properly leading to unpredictability in the architecture, development, and test	Led to cost overruns
	Architecture of the flight software - failure in thermal vacuum test in 2003	unpredictability -telemetry data was lost at a halt due to hardware design problems - in the development phase, the technologies were not mature enough to (TRL 6) to assess - in the test - architectural failures	flight software which had proven to be a complexity sponge as the design evolved with requirements clarity
	System Evaluation and Estimation of Resources to assess the software related technical risks	problems of risk returns	problems with quality attributes - maintainability, - testability, - interoperability, - scalability and - flexibility Failure in thermal vacuum testing (2007).
	untouchable by their own developers	There seems to be a standing army guarding this core system	NO EFFECT ON ACCIDENT
Test problems	testing of ground segment – not in accordance with the military standards by SEIT	-test team was still developing the test cases. -the test was not ready to accept the development inputs	test architecture was not taken into account before the beginning of the development
	TEMP was not properly developed	could not predict the technology risks undertaken by the development team	Technology risk
	One single Integrated T&E plan was developed – software maturity	the process was not utilized to the maximum ability	the test team lacked visibility into the development and risk analysis
	Potential failure paths were tested such as Fault Detection, Diagnostics and Recovery (FDDR) for Flight software		Failure in thermal vacuum test from 2003
	Simulated realistic environment “Test as fly and fly as you test” was practiced to avoid	the simulation tests had great failures due to workarounds implemented	Failure in thermal vacuum test from 2003
	IHC issued interim Authority to operate	inappropriately issued without proper investigation into the assurance capabilities	Lost confidence in the process

Context problems	testbeds were developed in parallel	difficult to distinguish between testbeds and flight software issues	-oversubscription of testbeds - lack of simulation resources - high-risk areas - the timing of stored programs - not tested
	system security features for HEO were found to be incomplete	HEO test data to be doubted	HEO capability to test, assess and support SBIRS was contested.
	Ground software development efforts were delayed due to database problems	development and test efforts had integration and total performance problems	faulty hardware and software design
	HEO P/L Single Board Computers had problems	latent defects were discovered in the manufacturing process of HEO in integration and test sequence	delays in the schedule -delays in the development had affected the test, which delayed further
	software development and control gyro reference assembly also failed during life testing in Increment 1	ground software problems resulted in a two-year slip	resulting in further postponements of delivery -the complexity of the technology
	As HEO 1 for Increment 2, the sensor delivery was delayed by a year		
	the delay was due to a first infrared sensor that had significant defects in flight software		
	first time integration of flight software was exercised in a new system	complexity in integration	remaining SBIRS program at risk -the technology was new so the progress in the process could not be determined.
	SBIRS High had problems in development integration	MR was depleting at a higher rate	cost and schedule variance
	GAO analysis reported challenges in assembly, integration, and test before the re-baseline	schedule delays and cost overruns	to meet the cost and schedule goals, some needed capabilities were deferred
	SBIRS High technology integration was a high risk	insufficient time	$2million of work was behind schedule for GEO -the schedule did not facilitate the delivery of the products leading to delays.
	Major design changes occurred to GEO late in design phase	technical problems found in testing	degraded sensor performance -enforcement of recommendations the failure that was repeated showing the inadequate enforcement
	SBIRS High critical design review with just 50% of design drawing was passed	HEO 1 had continuous changes to design due	cost and schedule escalations room for inappropriate creeps in requirements

the program did not invest enough time and resources in the basic systems engineering analysis		
Two late design changes were made to improve GEO satellites success- to improve operational reliability	the contractor was having difficulty assessing the specifications of the battery	schedule delays and increased cost - The design had to change after the clarity of requirements.
limited ability to identify all technical risks - up to date information was not available for thorough technical assessments - interrelationships among risks were not identified	-lack of an integrated management system - TSPR which removed the level of rigor in monitoring and assessing contractor capabilities - the inexperience of the staff - budget pressure, rushed environment and optimism about TSPR	contractors had projected optimistic claims - inaccurate and unrealistic cost performance index and schedule performance index - inadequate information about the progress - Risks not mitigated before the beginning of the development -the complexity of the technology -barriers to proceed were not defined
-IBR risks were rated moderate or lower -contractor risk assessments were either incomplete or over-optimistic	contractor's own incentive to bias the technical assessments	
TSPR approach had poor performance	reacquired in 2002 to regain the ability to properly oversee and assess contractor performance	Realignment was not done properly
COTS were assumed to be beneficial- deemed low risk without in-depth analysis	GEO bus underwent significant configuration changes and weight growth due to unique military requirements	-removal of military standards and specifications -quality control issues -serious technical problems such as HEO EMI problem
Contractor oversight was not appropriately managed	lack of disciplined approach to software development	inadequate coordination between cost and schedule functions
waivers were granted by software engineering process group to the software development process	meet the cost and schedule goals	increased the program risks -negligence to adhere to the process leading to problems in assessment processes -the process had many failures in the delivery mechanism
Waivers were approved for software design to be done in parallel with software specification	certain requirements have been rejected	rework in design and coding
the waiver was authorized for software unit integration testing to be done in parallel with formal unit testing	formal unit testing found problems that were not found in development unit testing	Rework - the extent to which change will cause damage to the mission goal
insufficient memory margin of the on-board satellite	waivers were granted	continued technical complexity challenges

	problems with acquisition policies governing basic system design	latent defect and process escapes	cost and schedule delays due to rework
	Technical risk assessments were underestimated due to immature technologies	compressed testing schedules	failures in meeting technical performance
	redesigns and reworks	schedule slips and cost growths	technical risks related to flight and ground software - first-time integration efforts were found to be of high risk
Context problems	HEO software development activities continued as a risk in GEO integration and test	Unexpectedly difficult in HEO integration.	
	variance at completion at $25.6 million and schedule variance of 32% whereas threshold variance was 5%	-integration testing and operations, -thermal vacuum test preparation -engineering rework such as Pointing and Control Assembly software was restructured to allow off-ramp option	
	tracking algorithms and software were not complete with hundreds of open defects and delayed qualifications	- accelerated ground software development - integration of GEO flight software which was high-risk effort did not start until August 2003	Schedule delays
	lack of coordination between ground software development and space		
	late delivery of database and inability of program office to reduce the length of time taken to certify data processed from GEO1		
	148 defective EMI frequencies- process failures, stringent requirements and the subcontractor did not implement the EMI control plan	delayed and increased the risk	Schedule delays -improper process -no clear understanding of individual contractor's responsibilities, so DOD did not bother to bring in new processes in place
	no contractor was clearly responsible for integrating HEO sensor with the host bus		
	-Hardware installation at the remote ground station, -legacy reporting system interfacing with MCS, -delays to start testing, -the requirement of testing of parallel operations prior to the declaration of IOC, -fault detection, and isolation problems	a significant delay in performance and reliability test	Schedule delays -lack of motivation to improve the efficiency of the process performance
	-Compressed timeline, -issues due to shared facilities at overseas relay ground stations, -delay in performance validation testing of increment 1 ground software,	increased risk and schedule delays and cost overruns	schedule delays and cost overruns -the schedule was very tight and did not incorporate the safety cushion

	- inadequate testbed design and scope, -simulation tests needed, -accelerated deployment of a low component in 2004, -significant improvement in SBIRS High requirements over DSP -inadequate HWIL testbeds		
	-the decline in the defense budget, - consolidation of the aerospace industry -growing competition for the fewer programs,	increased cost efficiency by transferring program responsibilities to contractors with less government oversight	Building lethargy -did not suit the process of development. -efficiency was not indicated as a requirement in the process
	-acquisition reform measures eliminated the usual cost and technical risk assessment data recording, -increased technological complexity -reduced acquisition workforce	challenged the knowledge to assess the technical and system engineering progress of the program	
	-Airforce instruction 99-101 “Developmental Test and Evaluation” not implemented - SMM tool was not used to track the progress of the program - all critical test plans and reports were not signed off	without effective management and oversight of development and testing, the program had the risk of repeating the problems identified during the program recertification	
Accident context	flight software in the single processor after the redesign	applications in a single core processor - non-distributed application architecture -grow in complexity and would be difficult to maintain	lower fail-safe mechanism and produces more heat
	GAO opted for redesign option	effectively design flight software architecture for efficiently performing software	proper mechanism to evaluate the redesign options
	GAO -sufficient evidence of the problem- oversight responsibility on Lockheed Martin during the trade study	proper oversight on the contractors	recommended this option of the redesign
	Airforce undertook requirements re-clarification effort along with the Lockheed Martin	recommended this redesign option	Airforce’s basis to recommend this redesign option is not clear.
	Program review board -assessing the architecture to the suitability of military requirements and recommends the design option	negligence on the part of the review team to have recommended the redesign option	negligence
	Independent review team - an independent assessment of design	did not do a rigorous analysis of the design presented for a redesign	oversight of the future performance problems this design would arise
	Joint Execution Team - flight software development - to conduct inch stone review, -Executive Program management	lack of disciplined process of Lockheed Martin and Airforce had limited control on SBIRS program	recommended separate program manager for flight software team

	-address weaknesses including Independent Program Assessment recommendations on technical baselines		
Accident Context	DOD authorized to proceed with development based on the redesign- Joint Execution Team	members were already players in the program	mitigate problems related to the original design of flight software
	DCMA monitoring the progress of software development – as delays would affect the launch	HEO software development was delayed - aggressive schedule -lack of understanding of the complexity of software tasks	higher defects - no clarity of the basis for assessment
	Flight software sensor - several defects in testing	HEO software development is among top ten program risks	Software development and integration, testing and assembly had problems including a sensor, pointing, and control assembly
	flight software development was significantly behind schedule	rework due to a higher amount of growing defects - did not monitor the development schedule and the progress	
	Air Force Requirements Oversight Council served to resolve operational requirements issues	The clarity of certain operational details required was not detailed well.	unplanned changes in development which led to increased complexity leading to growing defects
	DOD- Validation of system security features was incomplete	Data which were tested in HEO capability could not be trusted and the facility to perform tests were also questionable.	
	Designated Approval authority had inappropriately issued Interim Authority to Operate to Interim Highly Elliptical Orbit Capability	This violated Department of Defence Information Security Certification Authority by not ensuring the system security features were met by conducting security tests.	IHC data is considered incorrect as system security features such as availability, integrity, authentication, confidentiality were not validated against
	SEIT maintains requirement verification Ledger which is used in testing IPT maintains software requirements specification which uses Requirement Traceability and Management tool for test verification	Did not validate the tests properly to proceed to system test as the code reached the thermal vacuum testing and had major failures.	risks to be escalated as bigger to the stage of failure -is no check post to evaluate the credibility of progress
	AFOTEC- validating the operational effectiveness and checks the suitability of system in a cost-effective manner	the testing was not performed effectively to avoid major failure in thermal vacuum testing	
	The failure review board had assessed some functionalities as risky	the mitigation plans did not work leading to the risks carried further into later stages of the development cycle	

	SPO - there were many failures leading to major failure in the delivery of flight software in thermal vacuum testing - Waivers were requested by Lockheed Martin in software development process	-failures in the development process - These waivers evidently led to problems in design	-many design changes resulting from requirement specification clarification activity. - mounting defects in the formal unit testing due to the code not been tested in development testing -heavy rework

Table 2: Finding of this research

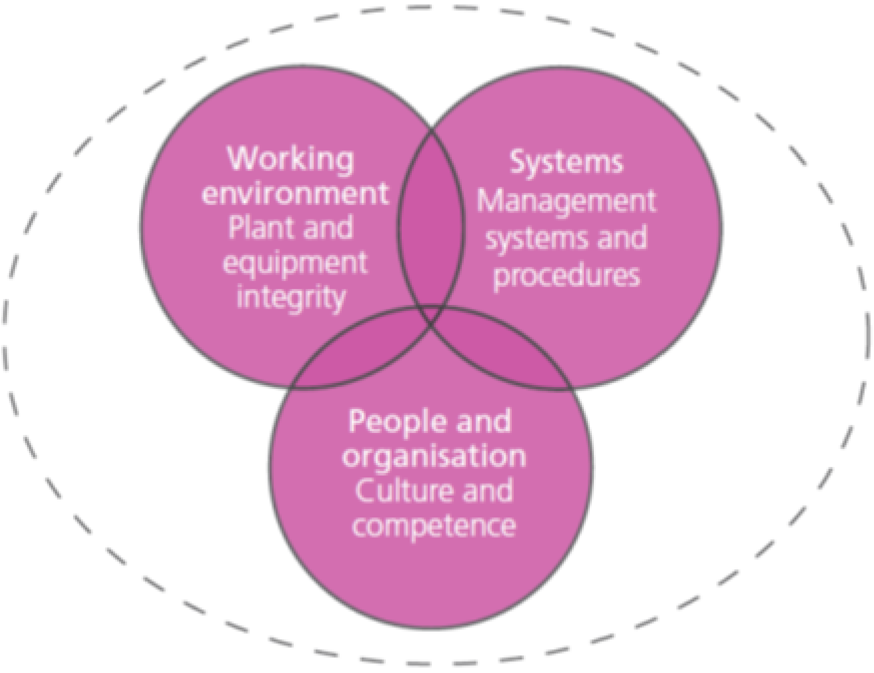
Chapter IV Recommendation of accident prevention

The context described in the above section is the moral sense of responsibility towards the self-image, it is an appendage to self-image by advising the action related to justified behavior, thereby improving self-esteem. This will, in turn, provide satisfaction to others which will provide evidence of supporting facts to honor the justice. It provides new meaning to the morale which benefits the welfare of the fellow men. This ruling of justice has to be executed which satisfies the wider community of the safer world. This is the context which is expected for successful execution of change.

In the case study, the context which is explicitly created for the deterrent is the cause of contributory negligence, the resulting context is in a state of restricted heroism with liabilities on individual contributions towards assumed success resulting in blaming the proximate cause as the failure. This factor of contributory negligence is an epidemic disease which is widely spreading that encourages individuals in such a context to develop double standards in conduct. That is one which is applied with a relaxed, subjective response which becomes an implicit standard. This environment of lethargy does influence the morale of the organizations involved to pick the pace only to see the excursion of the excitement of reaching the success.

The contextual factor is molded to the organizational needs in “The Health and Safety at workplace” which follows the principles of the organization’s culture improvement to have a safe environment. This has created a widespread acceptance of principles underlying the accident prevention techniques. Under this, the core is to promote positive culture through a holistic approach – the interaction between the working environment, equipment, systems and procedures and the people in the organization is considered[Institution of Occupational Health and Safety, 2015]. The culture of safety is promoted as shared values (what is important) and beliefs (how things work) that interact within an organization’s structure and control systems to produce behavioral standards (the way we do things around here)[Institution of Occupational Health and Safety, 2015]. Poor working conditions/culture would lead to accidents.

The challenge is to develop a positive organizational culture, as it is hard to change the attitudes and beliefs of a workforce by direct persuasion which has led to the development of “behavioral safety” approaches[Institution of Occupational Health and Safety, 2015]. The culture though develops slowly, the fundamental change does take time. The positive safety culture has three key aspects (*Figure 31*), working practices and rules for effectively controlling hazards, a positive attitude towards risk management and compliance with the control processes, the capacity to learn from accidents, near misses and safety performance indicators that bring about continual improvement[Institution of Occupational Health and Safety, 2015].

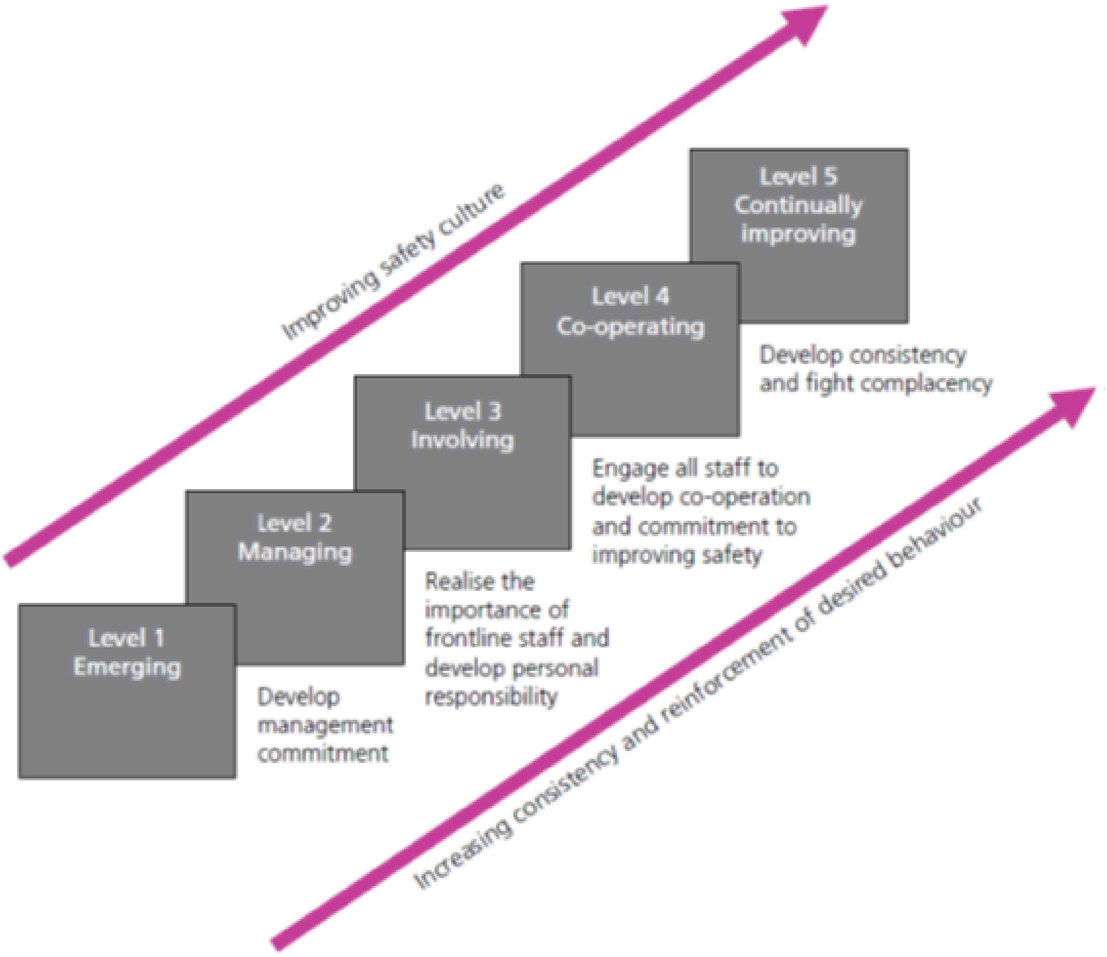

Working environment Plant and equipment integrity
Systems Management systems and procedures
People and organisation Culture and competence

Health and safety risk management: managing the risks associated with interactions between the working environment, the management systems, the organisation and its people

Figure 31 Positive safety culture[Institution of Occupational Health and Safety, 2015]

A maturity model for a culture[2] that can help to choose the right behavioral interventions for the organization is shown in *Figure 32* in five stages[Institution of Occupational Health and Safety, 2015].

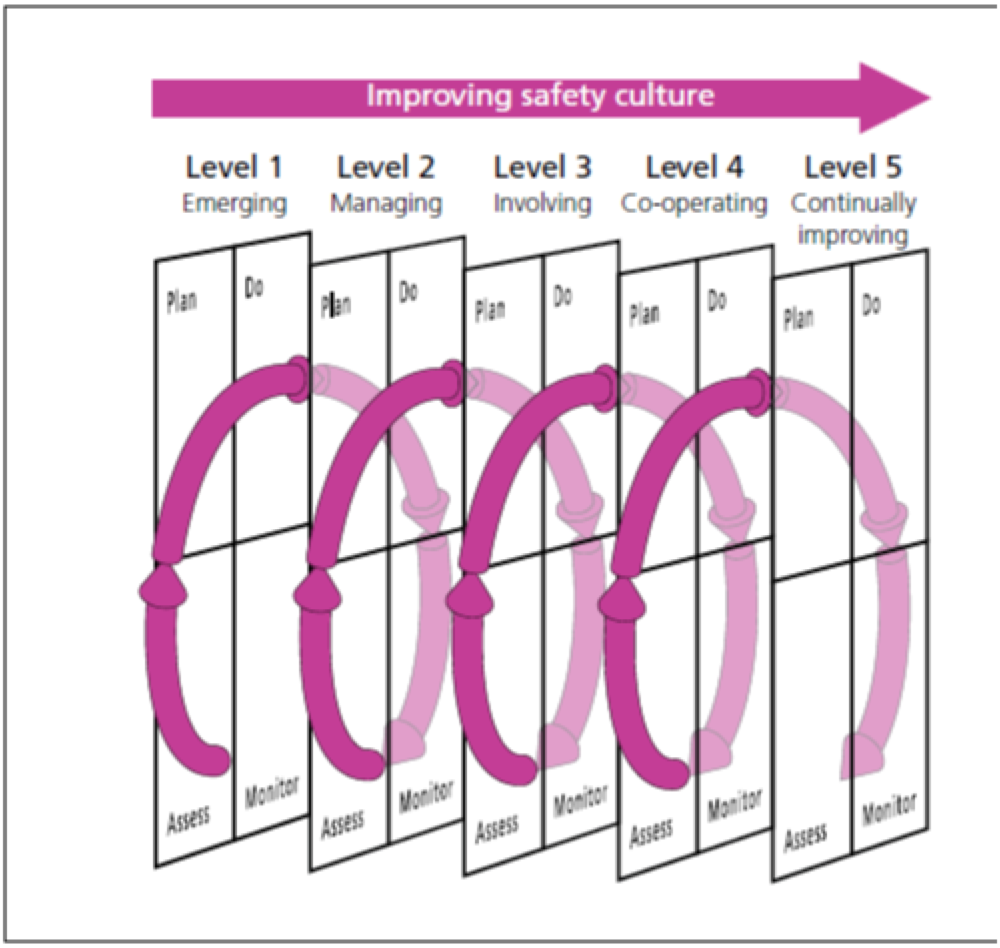
[2] ***Health and Safety Executive. Evaluating the effectiveness of the Health and Safety Executive’s Health and Safety Climate Survey Tool (RR042). HSE Books, 002.www.hse.gov.uk/research/rrpdf/ rr042.pdf.***



Safety Culture Maturity® model. © The Keil Centre 2001
;afety Culture Maturity is a registered trademark of The Keil Centre Ltd

Figure 32 Maturity Model for safety culture[Institution of Occupational Health and Safety, 2015]

The maturity model could be combined with principles of total quality management (*Figure 33*) to build safety culture change process by assessing the current level of maturity, developing a plan to move to next level, then implementing the plan, monitoring the implementation, reassessing the level of maturity to evaluate success and again identifying more actions[Institution of Occupational Health and Safety, 2015].



Safety culture change process

Source: Changing minds

Figure 33 Total Quality Management for change process[Institution of Occupational Health and Safety, 2015]

The Health and Safety procedures follow similar behavioral changes in the organization as the context of any change in the organization's ethos. The case study has accident cause as the context which is to be accommodated in order to have a successful sail of the SBIRS program. So this procedure of Health and Safety is recommended to ensure cultural change for the program to be more effective.

Chapter V Architecture Analysis

The architecture of having flight software placed in one processor is the accident that is detailed. The flight software which was originally designed to be placed in distributed application architecture and in two of the four processors in onboard satellite (*Figure 17*). The redesign of architecture which is to place the flight software in one processor (*Figure 18*) is to simplify the architecture than to have distributed application because it is the best fit with the system design.

The architectural issues here are:

1. Flight software on the ***single processor*.**

   Flight software on single processor means that one single processor has to bear the consequences of the heavy burden of applications for maintenance problems and issues of future development of flight software architecture. It was to solve the situation that of thermal vacuum testing where unexplained and unwarranted problems occurred due to architectural problems in the flight software, this solution of placing all applications on the single processor was adopted, to solve timing of stored programs, distribution of control between processors, and failure at the hardware interface level.

2. Flight software which is **not** on ***distributed application architecture***.

   Flight software which was not based on distributed application architecture was to be placed in single core processor. This would solve the problems by simplifying the architecture, developing more software, and increasing the robustness of the fault management system. This was seen as best fit with component and fault management and system design. Although applications were componentized into separate applications, they were not distributed into different processors. This distribution of applications is based on different architectural style as the “distributed application architecture”. This architecture takes the advantage of maximizing the performance of the processor by distributing the load of the processor to the processors which have the applications designed to accept such performance distribution.

Single processor: The processor is responsible for logical, computational and control activities of the computer. The processor can be of different types: a single processor and a multicore processor.

“If we do not write programs with a special focus on running on parallel cores, we will not get the advantage of multicores.”[Ghuman, 2016]

The multicore processors use two or more core to process the instructions. The advantages of using multicore processor are:

- Can execute multiple instructions by using multiple cores
- Can speed up the software which is designed for multi-core processors
- Has higher performance depending on a number of cores, frequency and software application to be executed.

“It is the single-core processors which are put together to make a multi-core processor.”[Ghuman, 2016]

The decision to place the flight software application on a single processor would not take the advantage of the multicore processor architecture. The original design was to use a multicore architecture which was altered to use single core processor after the thermal vacuum testing.

Distributed application architecture: Distributed applications are applications that are designed to run on multiple processors or computers. This architectural style advocates separation of concerns in the allocation of functionalities in the components. If the components are so designed to allocate the functionalities independently on separate processors, then the software is simple to understand and easier to develop and maintain[Fielding, 2000]. The general principle of simplicity is understood in terms of verifiability, usability, maintainability. Verifiability is the quality of software architecture to plan for the testing of functionalities, usability is the architectural quality which incorporates the reusability of the component in mind and maintainability is the architectural principle which focuses on reducing the defects and facilitates ease of maintenance.

Thus distributed application has the advantage of loose coupling of components which allows separate deployment into different processors that helps the applications to scale better and perform better.

Thus the decision to place flight software on a non-distributed application framework was a wrong decision as it had lost the advantages of scalability, maintainability, reusability, verifiability and performance loss as it is not taking the advantage of the multicore processor architecture.

The problem of architecture in this case study is multifold. The problem of hardware and software could be seen separately. The hardware side of the problem is to move from multicore architecture to single processor architecture. The software side of the problem is to move from distributed application architecture to legacy architectural style of placing all application components on a single processor (not distributed).

The flight software in the current state is on the distributed architecture taking the advantage of latest architectural principles such as Mission Data Systems and core flight software architecture.

Conclusion

The SBIRS case study was conducted based on the reports from GAO and DOD. These reports were for the period of 15 years. The analysis was done on decisions taken, statements from various other organizations involved that had influenced this accident. The reports from GAO and DOD did record the failure, this was not considered as an accident. So no further analysis was conducted thus far. This analysis was based on a failure which was not analyzed as the cause of the failures that followed pace.

At the onset of the reports from DOD and GAO, did point the failure in the architectural principle of moving from the distributed application architecture and multicore architecture to a non-distributed application architecture and single processor. Although this did seem as a wrong decision this analysis was convinced that NASA would not consider such an option. So a thorough analysis of the entire program was conducted to understand the circumstances in which such a decision had to be taken.

As the program had started in 1996, gathering evidence of all the decisions made was impossible. This analysis has collected all the relevant facts that could be related to the accident in this program. This information was validated for its correctness and completeness with the GAO Assistant Director who was involved in this program from the inception. The confirmation from the Assistant Director of GAO is in the Appendix (*Report 1*, *Report 2*).

All the failures were analyzed, all the organizations involved were identified and consequences of the failures were understood to create an understanding of the entire picture of the accident. As the failures were distributed laterally there was no clear line of hierarchy found in the structural analysis of the accident. So STAMP methodology was adopted for its loose adherence to hierarchical structures. This advantage was taken as a positive measure to adapt SBIRS accident to STAMP analysis. The failures were grouped under the organizations and organization's interactions were analyzed. The relationship between the failures was understood based on the consequences which escalated to form bigger failures.

STAMP was implemented in SBIRS accident, the STAMP methodology follows systematic elimination process of identifying the failures and organizations that had contributed to the interaction failures that STAMP prides in projecting it out as the consequence of the accident analysis. In the first phase of STAMP analysis, DOD was identified as the culprit, whereas the situation changed after analyzing the progressive phases of the accident to GAO and DOD being identified as organizations responsible for the accident. With the dynamic positioning of failures and cumulative positioning of the progressive accident, it was understood that problem lies with the context in which all organizations were operating. This realization was enlightening as the STAMP has brought out the true self of the accident cause.

The context as the cause of the accident had been of great influence in other accidents in various other organizations. This cause of the accident has never been considered as an accident ‘context’ from the onset of any accident analysis. So this analysis suggests context be considered as one of the factors responsible for methodical analysis of an accident. STAMP analysis is improved to consider context as one of the core principles of accident analysis.

The accidents which were reported were analyzed for prevention techniques, there were no techniques advised in any accident analysis this far for context as the cause of the accident. There is an attempt made to formulate a prevention technique by introducing the context as a factor to be controlled to prevent accidents in this case study. This could be done by using HSE process.

The accident in the case study is to determine the cause of failure, the decision taken by choosing a wrong architecture which was to place the flight software on one single processor that would not take advantage of both the distributed application architecture and a multicore processor. It is seen as the cause of the systemic failure than an isolated decision failure. The cause of this wrong decision was due to inertia built into the system over time to bear with anything that was thrown at it – a numbness which will anchor deep into the system.

As we have seen the wrong decision so taken was not the actual accident, the accident was built in over the years. This wrong decision is only the highlight of the accident. The accident was a loss of time and money in this program over 15 year’s period. The loss was unaccounted for, a hidden factor which no one can actually explain. So the context is chosen to be blamed here.

The STAMP, accident analysis process is commended for its procedural direction towards this outcome. The outcome though was shocking as all the organizations and all the interactions had resulted in such a realization which stands still as a mystery for many decades of the guilt of being hostile in living with it than to have taken preventive measures to avoid future accidents. As there seems to be no preventive action that could be suggested even at this stage which was successful in preventing accidents with context as a causal factor.

This analysis has contributed towards analyzing the accidents that were not considered as accidents as the end result had been successful. Such accidents go unnoticed but would have contributed towards the failure of the mission by delays and cost overruns. It is required now in this era to understand progressive phases of accidents before it culminates to be an accident. This case study has analyzed the progressive accident to learn from it so that the end result of the program could be altered. This case study has analyzed the accident for the period of 15 years. As the organizations are procedural and systematic, the interactions are structured in the military environment, this case study was chosen for its merits. STAMP methodology was used in analyzing the case study for understanding the progressive accidents to identify prevention techniques before the accident disrupts the mission. An improvement in the STAMP methodology was suggested and a process to suit the STAMP improvement was suggested for preventive accident analysis.

Recommendations for Future Work

The effectiveness of software depends on factors such as an underlying formal model (ambiguity can be avoided); compositional capabilities (safety properties of components); a system modelling approach; sufficient effective power to capture common failure scenarios; automation support for accident analysis[Wu and Kelly, 2005].

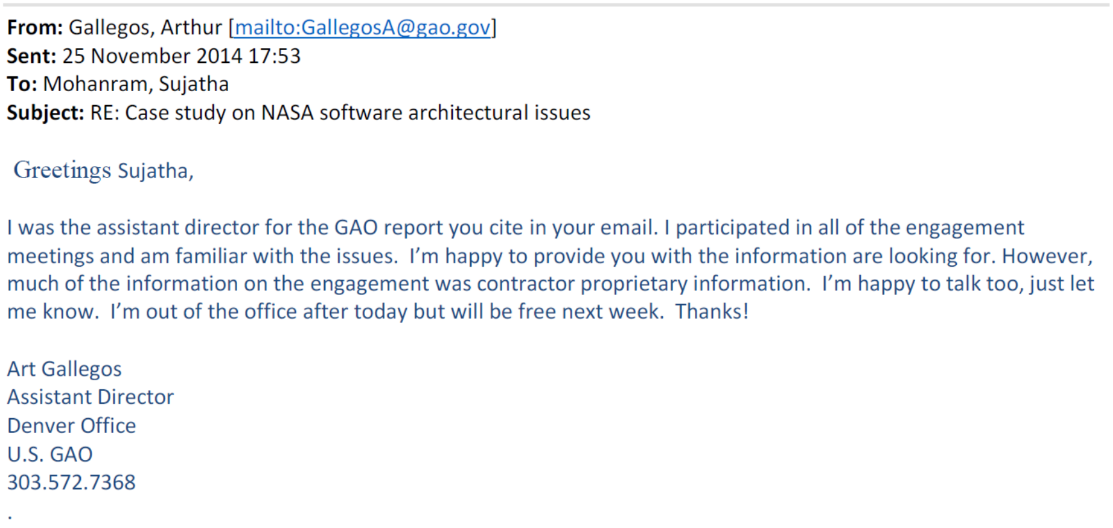
"Software architecture addresses software issues such as flexibility and development costs by providing an abstract model of a system in terms of software elements that have externally visible properties and their interactions"[Wu and Kelly, 2005].

Failure modelling of software architecture was attempted by using Communicating Sequential Process (CSP) language[Wu and Kelly, 2005]. In CSP, a basic unit is an event, and the building block is a process representing a pattern of event sequences[Wu and Kelly, 2005]. The failure behavior considers processes of each component[Wu and Kelly, 2005]. The architectural views are modelled in this failure modelling framework to isolate functional failures and hardware failures[Wu and Kelly, 2005]. The main focus of failure modeling would be the definition of failure behavior of elementary components[Wu and Kelly, 2005].

There is a realization that the same architectural problems are repeated over and over again in many organizations. All organizations have their own evolution of architecture to solve their problems. It is thought that there should be common lessons learned to build a framework incorporating the common evolution of architecture. The next phase would be to standardize this evolved framework. It is proposed that a "***blueprint*** for ***software architecture***" will bring common lessons learned into a framework that could be standardized. Hoping to progress towards recommending an appropriate model in this in future. Daniel Dvorak chief technologist in NASA has invited in the flight software complexity project where this idea would be progressed.

The cause of the accident is the "context" in which the organizations function. The accident analysis is undertaken to understand and prevent the accidents from repeating. In this case study, it was analyzed that the context is an additional factor that should be taken into consideration. The prevention of accident due to context is still unknown. This could be considered as future work. It has been progressed by feeding the context to the organizations. This idea will be progressed in Daniel Dvorak's, NASA - flight software complexity project.

Appendix



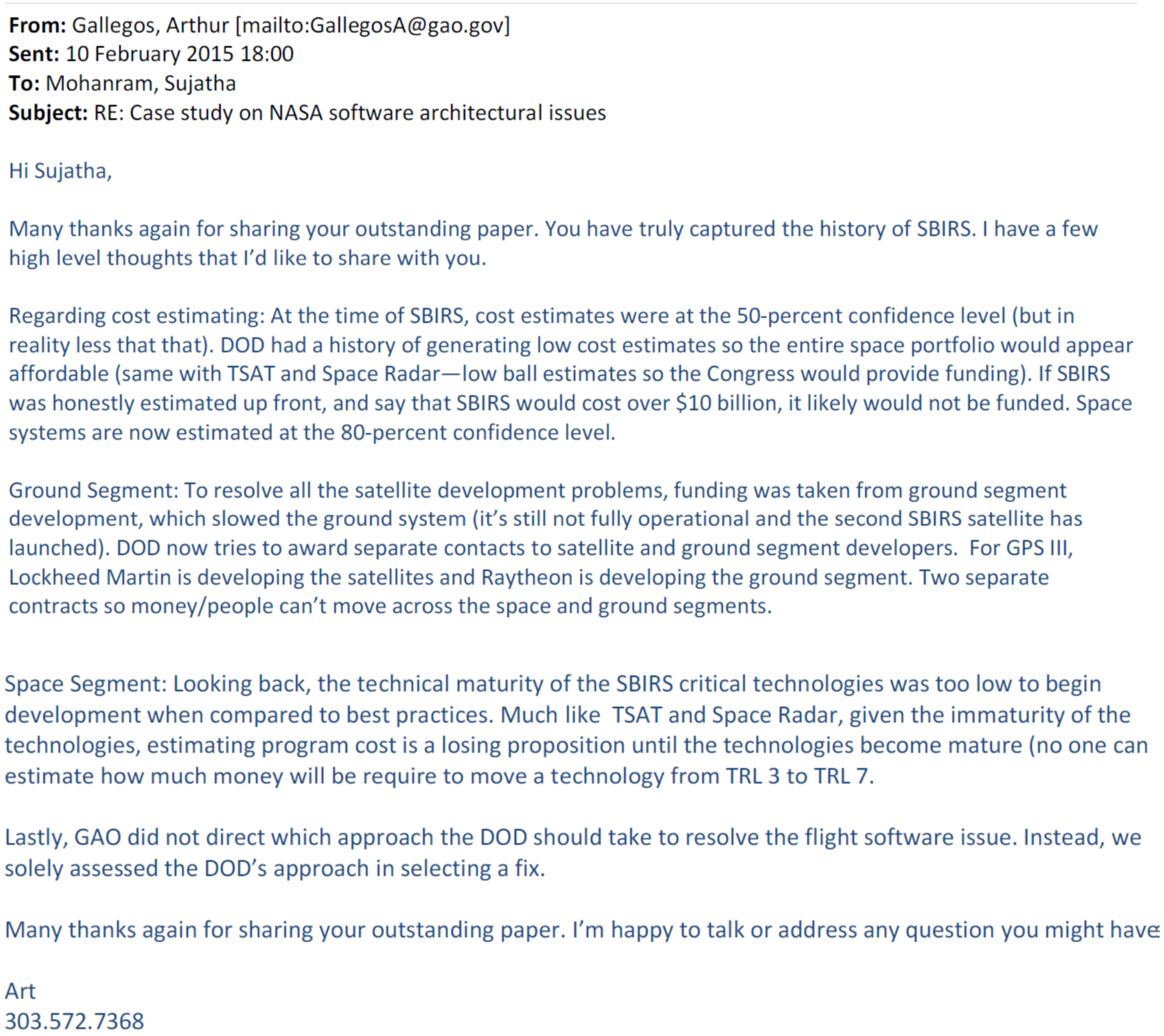
From: Gallegos, Arthur [mailto:GallegosA@gao.gov]
Sent: 25 November 2014 17:53
To: Mohanram, Sujatha
Subject: RE: Case study on NASA software architectural issues

Greetings Sujatha,

I was the assistant director for the GAO report you cite in your email. I participated in all of the engagement meetings and am familiar with the issues. I'm happy to provide you with the information are looking for. However, much of the information on the engagement was contractor proprietary information. I'm happy to talk too, just let me know. I'm out of the office after today but will be free next week. Thanks!

Art Gallegos
Assistant Director
Denver Office
U.S. GAO
303.572.7368
.

Report 1: GAO Assistant Director, familiarity with the SBIRS program



From: Gallegos, Arthur [mailto:GallegosA@gao.gov]
Sent: 10 February 2015 18:00
To: Mohanram, Sujatha
Subject: RE: Case study on NASA software architectural issues

Hi Sujatha,

Many thanks again for sharing your outstanding paper. You have truly captured the history of SBIRS. I have a few high level thoughts that I'd like to share with you.

Regarding cost estimating: At the time of SBIRS, cost estimates were at the 50-percent confidence level (but in reality less that that). DOD had a history of generating low cost estimates so the entire space portfolio would appear affordable (same with TSAT and Space Radar—low ball estimates so the Congress would provide funding). If SBIRS was honestly estimated up front, and say that SBIRS would cost over $10 billion, it likely would not be funded. Space systems are now estimated at the 80-percent confidence level.

Ground Segment: To resolve all the satellite development problems, funding was taken from ground segment development, which slowed the ground system (it's still not fully operational and the second SBIRS satellite has launched). DOD now tries to award separate contacts to satellite and ground segment developers. For GPS III, Lockheed Martin is developing the satellites and Raytheon is developing the ground segment. Two separate contracts so money/people can't move across the space and ground segments.

Space Segment: Looking back, the technical maturity of the SBIRS critical technologies was too low to begin development when compared to best practices. Much like TSAT and Space Radar, given the immaturity of the technologies, estimating program cost is a losing proposition until the technologies become mature (no one can estimate how much money will be require to move a technology from TRL 3 to TRL 7.

Lastly, GAO did not direct which approach the DOD should take to resolve the flight software issue. Instead, we solely assessed the DOD's approach in selecting a fix.

Many thanks again for sharing your outstanding paper. I'm happy to talk or address any question you might have

Art
303.572.7368

Report 2 : Assistant Director, GAO- Confirmation statement on SBIRS project

Reference

DOD. *Development Testing of Space Based Infrared System Mission- Critical Software*. [*Online*]. *Office of the Inspector General*, 2003. Available at:http://www.dodig.mil/Audit/reports/fy04/04-022.pdf.

Dvorak, Daniel L. *NASA Study on Flight Software Complexity*. [*Online*]. *NASA Ofice of Chief Engineer*, 2009. Available at:doi:10.2514/6.2009-1882.

Evans, Ben. *Atlas V Carrying the SBIRS GEO - 2 Satellite Launches Successfully*. [*Online*]. *Wired4Space*, no. March, 2013 pp. 1–9. Available at:http://www.wired4space.com/space/atlas-v-with-sbirs-geo-2-satellite-ready-for-launch.

Feller, PH, D Gluch, and K Woodham. *Case Study: Model-Based Analysis of the Mission Data System Reference Architecture*. [*Online*], no. May, 2010. Available at:http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA528582.

Fielding, Roy Thomas. *Architectural Styles and the Design of Network-Based Software Architectures*. [*Online*]. UNIVERSITY OF CALIFORNIA, IRVINE, 2000. Available at:https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf.

GAO. *Despite Restructuring , SBIRS High Program Remains at Risk of Cost and Schedule Overruns*. [*Online*], 2003. Available at:www.gao.gov/cgi-bin/getrpt?GAO-04-48.

GAO. *Space Based Infrared System High Program and Its Alternative*. [*Online*], 2007. Available at:http://www.gao.gov/products/GAO-07-1088R.

GAO. *DOD ' S Goals for Resolving Space Based Infrared System Software Problems Are Ambitious*. [*Online*], 2008. Available at:http://www.gao.gov/new.items/d081073.pdf.

Ghuman, Sukhdev Singh. *Comparison of Single-Core and Multi-Core Processor*. [*Online*] 6 (6), 2016 pp. 423–24.

Institution of Occupational Health and Safety. *Promoting a Positive Culture*. [*Online*].

IOSH, no. May, 2015 pp. 1–16. Available at:www.iosh.co.uk/positiveculture.

Jay A. Moody, Major. *Achieving Affordable Operational Requirements on the Space Based Infrared System (SBIRS) Program: A Model for Warfighter and Acquision Success.* [*Online*]. Air Command and Staff College, 1997. Available at:http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA397934.

Leveson, Nancy. *A New Accident Model for Engineering Safer Systems*. [*Online*]. *Safety Science* 42 (4), 2004 pp. 237–70. Available at:doi:10.1016/S0925-7535(03)00047-X.

Leveson, Nancy. *A Safer World*. [*Online*]. *Chemistry and Industry (London)* 77 (8), 2011 pp. 42–45. Available at:doi:10.1002/cind.7708-14.x.

Leveson, Nancy, Mirna Daouk, Nicolas Dulac, and Karen Marais. *Applying STAMP in Accident Analysis*. [*Online*]. *NASA Conference Publication*, 2003, 177–98. Available at:http://esd.mit.edu/WPS/esd-wp-2003-02.pdf.

LockheedMartin. *First SBIRS Satellite Exceeding Performance Expectations After One Year on Orbit*. [*Online*]. *Lockheed Martin Media*, no. June 5, 2012. Available at:http://www.lockheedmartin.co.uk/us/news/press-releases/2012/june/0605-ss-sbirs.html.

Mccaney, Kevin. *Air Force Awards $ 1 . 86B Contract for next Two SBIRS Satellites.* [*Online*]. *C4ISR*, no. 25 June, 2014 pp. 1–7. Available at:https://defensesystems.com/articles/2014/06/25/air-force-lockheed-sbris-satellite-contract.aspx.

Mccomas, David. *NASA / GSFC ' S Flight Software Core Flight System*. [*Online*], 2012. Available at:http://docplayer.net/41145319-Nasa-gsfc-s-flight-software-core-flight-system.html.

Office of the Secretary of Defense. *Status of the Space Based Infrared System Program.* [*Online*]. *Defense and Intelligence Committees of the Congress of the United States*, 2005. Available at:http://nsarchive.gwu.edu/NSAEBB/NSAEBB235/42.pdf.

Qureshi, Zahid H. *A Review of Accident Modelling Approaches for Complex Critical*

Sociotechnical Systems. [*Online*]. *12th Australian Workshop on Safety Related Programmable Systems (SCS'07), Adelaide* 86, 2008 pp. 47–59. Available at:http://www.dtic.mil/get-tr-doc/pdf?AD=ADA482543.

SBIRS. *Space-Based Infrared System (SBIRS)*. [*Online*]. *Airforce Programs*, no. December, 2013 pp. 297–98. Available at:http://www.dote.osd.mil/pub/reports/FY2013/pdf/af/2013sbirs.pdf.

Song, Yao. *Applying System-Theoretic Accident Model and Processes (STAMP) to Hazard Analysis*. [*Online*], 2012. Available at:https://macsphere.mcmaster.ca/bitstream/11375/11867/1/fulltext.pdf.

Spence, Author Rob. *Thought Leadership The Importance of Organisational Leadership Thought Leadership The Importance of Organisational Leadership*. [*Online*], no. 6, 2009. Available at:https://www.bainessimmons.com/wp-content/uploads/importance-of-organisational-leadership.pdf.

Toolbook, The Quality, Colossal Iceberg, Breaks Away We, Iceberg Larsen, and Pope Francis. *Food Quality Mgmt . System Food Quality Mgmt . System*. [*Online*], 2017, 1–3. Available at:http://www.syque.com/quality_tools/toolbook/toolbook.htm.

Wu, Weihang, and Tim Kelly. *Failure Modelling in Software Architecture Design for Safety*. [*Online*]. *Proceedings of the 2005 Workshop on Architecting Dependable Systems*, 2005, 1–7. Available at:doi:10.1145/1082983.1083222.

Younossi, Obaid, Mark A. Lorell, Kevin Brancato, Cynthia R. Cook, Mel Eisman, Bernard Fox, John Graser, Yool Kim, and Robert S. Leonard. *Improving the Cost Estimation of Space Systems: Past Lessons and Future Recommendations*. [*Online*]. RAND Project Airforce, 2008. Available at:http://www.rand.org/content/dam/rand/pubs/monographs/2008/RAND_MG690.pdf

Zacks. *Lockheed Martin Clinches $1.9B GEO-5 and GEO-6 Satellite Contract*. [*Online*]. *NASDAQ*, no. June 25, 2014 pp. 1–3. Available at:http://m.nasdaq.com/article/lockheed-martin-clinches-19b-geo-5-and-geo-6-satellite-contract---analyst-blog-cm365006.