ASSESSMENT OF ENGINEERING METHODOLOGIES FOR INCREASING CUBESAT MISSION SUCCESS RATES

A Dissertation
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Abdulaziz Alanazi

In Partial Fulfillment of the Requirements
for the Degree of
DOCTOR OF PHILOSOPHY

Major Department:
Computer Science

February 2020

Fargo, North Dakota

NORTH DAKOTA STATE UNIVERSITY

Graduate School

Title

ASSESSMENT OF ENGINEERING METHODOLOGIES FOR INCREASING CUBESAT MISSION SUCCESS RATES

By

Abdulaziz Alanazi

The supervisory committee certifies that this dissertation complies with North Dakota State University's regulations and meets the accepted standards for the degree of

DOCTOR OF PHILOSOPHY

SUPERVISORY COMMITTEE:

Dr. Jeremy Straub
Chair

Dr. Kenneth Magel

Dr. Hassan Reza

Dr. Derek Lehmberg

Approved:

02/19/2020
Date

Dr. Kendall Nygard
Department Chair

ABSTRACT

In the last twenty years, CubeSat Systems have gained popularity in educational institutions and commercial industries. CubeSats have attracted educators and manufacturers due to their ability to be quickly produced and their low cost, and small sizes and masses. However, while developers can swiftly design and build their CubeSats, with a team of students from different disciplines using COTS parts, this does not guarantee that the CubeSat mission will be successful. Statistics show that mission failure is frequent. For example, out of 270 "university-class" CubeSats, 139 failed in their mission between 2002 and 2016 [1]. Statistics also show that the average failure rate of CubeSat missions is higher in academic and research institutions than in commercial or government organizations.

Reasons for failure include power issues, mechanical, communications and system design issues. Some researchers have suggested that the problem lies within the design and development process itself, in that CubeSat developers mainly focus on system and component level designs, while neglecting requirements elicitation and other key system engineering activities [2]. To increase the success rate of CubeSat missions, systems engineering steps and processes need to be implemented in the development cycle. Using these processes can also help CubeSat designs and systems to become more secure, reusable, and modular.

This research identifies multiple independent variables and measures their effectiveness for driving CubeSat systems' mission success. It seeks to increase the CubeSat mission success rate by developing systems engineering methodologies and tools. It also evaluates the benefits of applying systems engineering methodologies and practices, which can be applied at different stages of CubeSat project lifecycle and across different CubeSat missions.

ACKNOWLEDGEMENTS

To my parents (God bless their souls), thank you for encouraging me in all my pursuits and inspiring me to follow my dreams. To my siblings, my deepest appreciation for your emotional and financial support. I always knew that you believed in me and wanted the best for me. To my lovely wife and children, this journey would not have been possible without your love and support. I am extremely grateful to have you in my life. I owe a debt of gratitude to my advisor, Dr. Jeremy Straub, for his time, support, and guidance throughout this four-year journey. I would like to extend my deepest gratitude to my committee members and the following academic and industry professionals who helped me with their valuable feedback and suggestions: David Kaslow (INCOSE), Dr. David Sternberg (NASA JPL), Dr. Hassan Alsuhabi and Curt Doetkott (Department of Statistics, NDSU), Dr. Mohammed Alziyadi (Department of Physics, NDSU), my colleagues, Dr. Ahmad Bujalawi and Dr. Andrew Jones (Department of Computer Science, NDSU), and the CubeSat community.

DEDICATION

This dissertation is dedicated to the person who taught me that education is the key to intellect.

My loving brother, Fayadh (Abu Salah).

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION[1]

Cube-shaped miniature satellites, called CubeSats, are used for space exploration and research. CubeSat designs are comprised of multiple cubic units of 10 cm × 10 cm × 11.35 cm and have a mass of approximately 1.33 kg per each cubic unit (1U) [1]. CubeSats have structural and electronic components. They are typically developed using commercial off the shelf (COTS) components [1]. Many CubeSats are deployed from the International Space Station. They are also commonly launched on rockets as secondary payloads. CubeSats are considered one of the most efficient spacecraft [1]. They can carry out various space missions for commercial, scientific, government and military purposes and are used for Earth observations and amateur radio [1]. Most CubeSats are designed, and intended, to be put in the low-Earth orbit (LEO) with an altitude between 160-2,000 km above the Earth's surface [2]; however, some CubeSats are now being built for deep space applications [2].

The idea of building a small cube-shaped spacecraft was initiated by Bob Twigg and Jordi Puig-Suari in 1999 [3]. CubeSats were designed to be used for educational purposes, to help students to become acquainted with the space environment. CubeSats have taught students and researchers from the science, technology, engineering and math (STEM) disciplines, about the architecture, development and operations of a real functioning spacecraft system [4].

CubeSats are not only used for educational purposes, but also for scientific, governmental, and commercial purposes. Since 2014, the use of CubeSats to perform industrial work, and for commercial projects, has increased [5]. Their use has also grown in education. Data collected by Swartwout shows that between 2010 to 2016, 325 1U CubeSats were launched from 166 educational institutions in 47 countries [5]. The number of launched CubeSats has dramatically increased in the last eight years. As of January 2020, more than 1,200 CubeSats have been launched and the number is expected to grow [6]. The number of launched CubeSats since 2003 is depicted in Figure 1.1 [6].

[1]Based on A. Alanazi and J. Straub. 2019. Engineering Methodology for Student-Driven CubeSats. Aerospace, Vol. 6, No. 5.

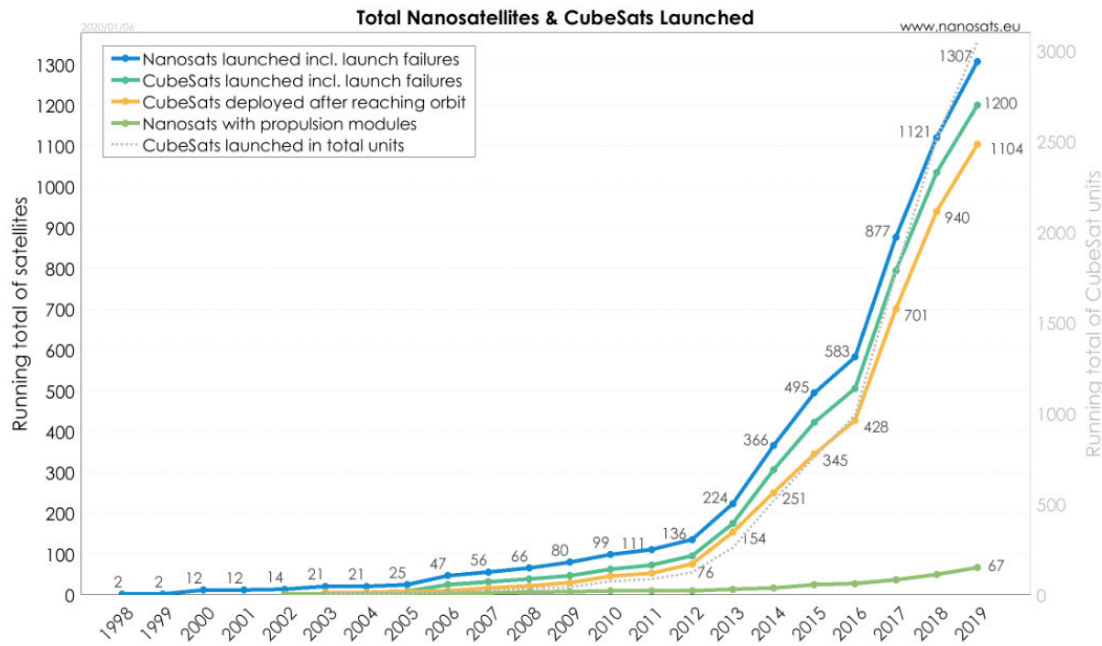


Figure 1.1. Total number of launched CubeSats, including launch failures, from 2003 to 2019 [6].

1.1. CubeSats as Part of the Space Ecosystem

The wide use of CubeSats has been enabled by standardization. CubeSat systems include an important part known as a dispenser. This is the interface between the launch vehicle and the CubeSat. The dispenser is responsible for the protection of the CubeSat and the launch vehicle, its primary payload, and other secondary payloads during the launch period. The dispenser releases the CubeSat into space at the proper time. Launch vehicles have the dispenser integrated into them and take the CubeSat into orbit [7]. Developing a CubeSat system in recent years has become easier for educational institutions, industries, and even hobbyists. However, just designing and building a CubeSat does not guarantee that it will make it into orbit. In the United States, like in other countries, there are certain licenses and regulations that cover a CubeSat system's activities that need to be obtained prior to launch [8]. The United States' CubeSat-relevant agencies are listed in Table 1.1. There are also main metrics used to classify a spacecraft by mass which are presented in Table 1.2.

Table 1.1. Regulatory agencies.

Agency	Role
Federal Communications Commission (FCC)	Regulates radio frequencies (RF) and provides orbital debris risk analyses [9]
National Aeronautics and Space Administration (NASA)	Funds missions and provides technical knowledge and launch services for education [10]
National Oceanic and Atmospheric Administration (NOAA)	Regulates remote sensing, if a CubeSat includes imaging payload [11]
National Science Foundation (NSF)	Funds missions [12]
U.S. Air Force	Funds missions and provides technical knowledge [13]

Table 1.2. Satellites mass classifications [14].

Type	Mass
Large Satellites	>1,000 kg.
Medium Satellites	Between 500-1,000 kg.
Small Satellites	<500 kg.
Minisatellites	100-500 kg.
Microsatellites	10-100 kg.
Nanosatellites	1-10 kg.

1.2. CubeSat Components and Specifications.

CubeSats have a number of standard core components including an antenna, a radio transmitter that is used for uplink and downlink, an on-board computer (OBC), sensors, and a power system that will typically include a battery [7]. These components are standard in most spacecraft, not just CubeSats. CubeSats have structural differences for different missions. It is essential that structural components be volume efficient. because of this, in many cases, structural components serve as the primary component for thermal management [15]. Additionally, structures must provide needed radiation shielding and serve, if required, as a pressure containment vessel [15].

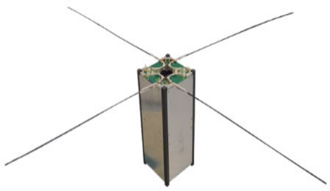
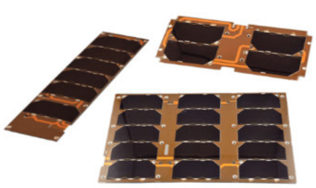
Components of CubeSats have included solar sensors, solar panels, electron canons, an attitude determination and control system (ADCS), an electromagnet, a command and data handling system (CDHS), an electrical solar wind sale, a tether reel motor, an electrical power system, a communication system, an antenna for downlink, an antenna for uplink, an onboard camera, a tether end mass, an access port, a frame and side panels [16]. Some of these components are highly standardized, while others have been developed to support specific mission needs. A list of standard CubeSat main components and what they do is included in 1.3.

Table 1.3. CubeSat components and specifications.

Type	**Description**
On-board computer [17] 	The On-board computer (OBC) commands the spacecraft and sends orders to different subsystems. It also can receive and store information for the CubeSat.
Attitude Determination and Control System (ADCS) [17] 	The attitude determination and control system (ADCS) is a subsystem of a CubeSat that provides stability and pointing capabilities. These are needed for many missions' antennas and payloads.

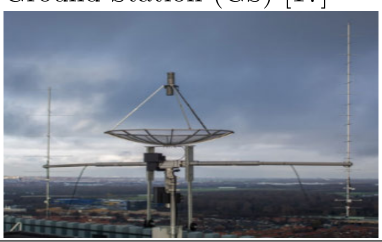
Continued on Next Page...

Table 1.3. CubeSat components and specifications (Continued).

Type	Description
Antenna [17] 	The antenna is used for transmitting and receiving data. Antennas are designed for specific frequency configuration needs.
Electrical Power System (EPS) [17] 	The EPS receives power from the solar cells, stores it and provides power to other components of the CubeSat system.
Transceiver [17] 	The transceiver is a radio for ground station and craft to craft communications. Transceivers can be half or full duplex and can operate in a variety of commercial and amateur bands across the VHF/UHF frequencies and other parts of the spectrum.
Solar Panels [17] 	Solar panels generate power for the spacecraft. Some solar panel systems contain thermal and sun sensors.

Continued on Next Page...

Table 1.3. CubeSat components and specifications (Continued).

Type	**Description**
Chassis [17] 	The chassis is the primary structure of the spacecraft. They typically are made with aluminum alloys 7075, 6061 and 5052. They need to be anodized or otherwise treated to prevent cold welding to the launcher.
Ground Station (GS) [17] 	The ground station is used to communicate with the spacecraft. Some can autonomously track selected satellites using a steerable antenna system.

CubeSat subsystems are typically designed and developed separately before being combined with all the other components. Some spacecraft may combine COTS components. Because of this, software developers can design modular, system-specific software and code and test it. Then the components and the supporting software can be integrated together and tested. This allows developers to divide up this process into small more manageable iterations. A block diagram of CubeSat subsystems (in reference to the OBC), and a typical 1U CubeSat diagram are shown in Figures 1.2 and 1.3.

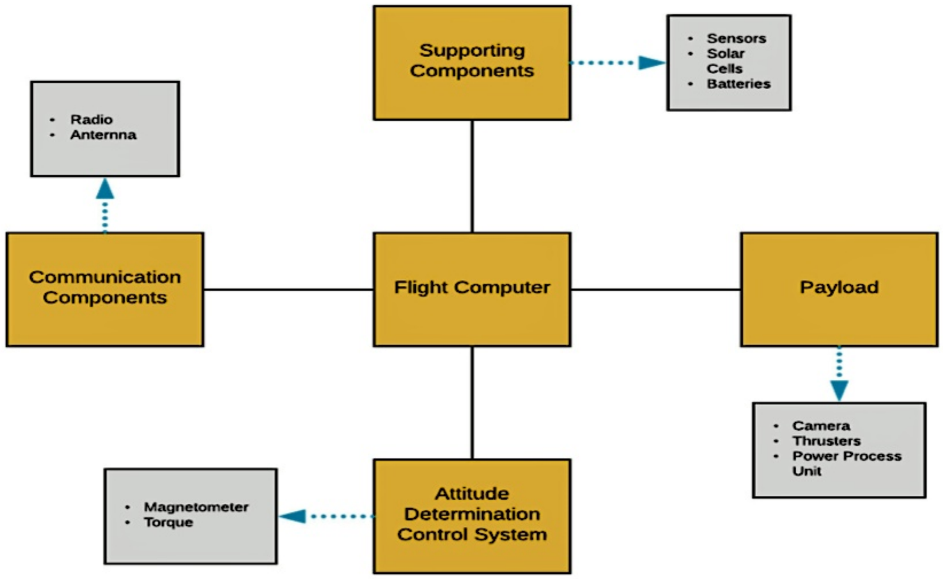


Figure 1.2. A block diagram of a CubeSat subsystem's components [18].

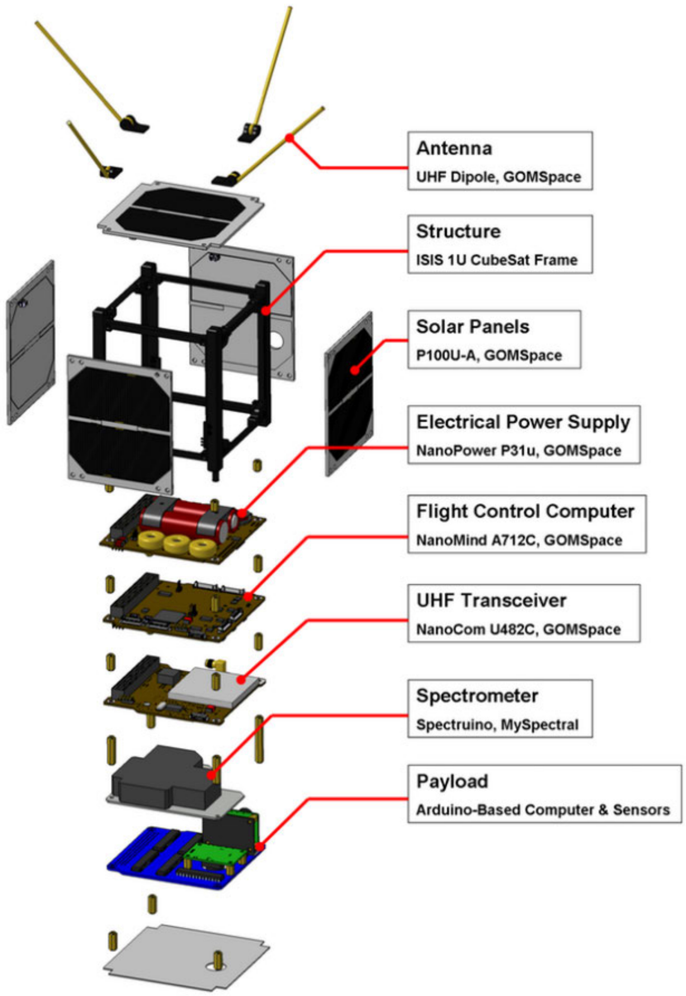


Figure 1.3. Structure and components for an example 1U CubeSat [19].

1.3. Communications with Ground Stations (GS)

CubeSats stay in orbit because of the balance between gravitational force and centrifugal force. According to Newton's theory, any two masses are attracted to each other by a gravitational force of magnitude [14]. This is explained in the equation of force balance:

$$\frac{GMm}{r^2} = mr\omega^2, \tag{1.1}$$

where (G) is the gravitational constant, (M) is the mass of first object, (m) is the mass of second object, (r) is the distance between Earth and the satellite, and (ω) is the angular velocity.

When a CubeSat is in space, it needs a ground station for communications. There are three main components of communications in CubeSats systems: the ground station, the onboard computer, and the onboard radio. A communication channels is established between the radio and the ground station. The radios and the antenna are the main components of the communications process [7]. Designing the communications system requires consideration of orbital characteristics, power requirements, hardware capabilities and the modulation parameters for the transmission of signals [20].

There are two channels between the ground station and a CubeSat: one is uplink (from the ground station to the satellite); the second is downlink (from the satellite to the ground station). Uplink sends commands to the CubeSat. The downlink is used to send data to the GS where it may be further processes, interpreted, and/or stored. For many sensing missions, a key role of the GS is to store all the images captured by the satellite for future use [20]. Mission planners typically use software tools to plan mission activities and define the tasks that must be performed. In some cases, software is used to continuously monitor the health of the spacecraft [21]. Managing a CubeSat constellation using COTS software tools facilitates mission management using the very limited bandwidth between the GS and the satellite.

To help CubeSats function and operate in orbit, ground stations needed to have this functionality. Ground Stations are equipped with COTS hardware and software that help in tracking and commanding operating CubeSats. Some CubeSats utilize downlink and uplink frequencies that are in the amateur radio bands [15]. Additionally, GS systems are responsible for commanding the payload, controlling CubeSat subsystems, and processing telemetry [14].

To track and control a CubeSat, tracking software is needed for this process, along with backend receivers and antenna with a positioner. In some cases, this some software also supports commanding the CubeSat and decoding the beacon [22].

1.4. CubeSat Engineering

What makes CubeSats different than traditional large satellites is their small size. This changes design considerations dramatically. The smaller scale of a CubeSat leads to a lower number of components. This may reduce the probability of component failure; however, it also removes redundancy. CubeSats also, in many cases, do not need complex thermal and structural designs, because they use coatings and insulation to minimize dark/light temperature variations [14]. Cube-

Sats have a large surface area to volume ratio [14]. They typically manage their internal temperature without reading electric heating. CubeSats are simple in design, as compared to large satellites. They have smaller instruments, perform less data collection, and the amount of information needed to downlink is lower than larger spacecraft. Power is typically constrained [14]. Thus, it is critical to work with payload developers to scrutinize the uses of power and minimize them and to maximize power production. It is also important to choose orbits and attitudes well suited to the production of solar energy.

The team sizes typically used for developing CubeSats are also different. CubeSat teams are comparatively small: 6-15 people versus thousands of people who must somehow be managed in contributing to a major program [23]. An organization of 15 or fewer creating a system in 2 years or less is fundamentally different than a large program [23]. Every person in the program typically communicates with every other team member [24]. Optimizations, in use of on-board resources (power, mass, volume, viewing angle, computing power, downlink bandwidth), and in use of personnel, are global instead of limited to a subsystem development team which works to a set of interfaces fixed during the systems engineering and planning process.

Documentation is reduced to only that which is produced in the process of engineering. Most of the information on how the system is designed and integrated is shared among the team members, rather than codified in writing and diagrams [24]. In fact, despite the documentation created in larger systems, reliability and manufacturability of these systems still relies critically on staff communications and continuity [24]. Instead of formal reviews, which greatly increase the workload on the small team and may exceed team capabilities, informal processes are used [24]. Technical interchange meetings, specific to subsystems, and at the systems level (without delving into subsystems details), are held around a conference table with limited outside reviewers and only those team members directly concerned using actual design documents, drawings, spreadsheets, simulations and calculations [24].

Rather than relying on detailed analysis, some small programs emphasize testing [24]. For all progress, it is imperative to manage the analysis, design, development, and integration phases to conserve time, personnel, and financial resources for testing. Most subsystems are only unit tested at the benchtop level, not in thermal vacuum chambers and other special facilities, because the entire micro spacecraft is smaller and less complex than the typical conventional spacecraft subsystem [25]. Thus, the manager accepts the risk that a subsystem might fail in systems level testing, in exchange for the savings in time and money realized by moving to systems-level testing sooner.

A CubeSat project's life cycle includes multiple phases and steps. However, some of these steps may be less critical than others and can be skipped, in some cases, or can receive less time for development. The phases of a project's life cycle can be divided into small iterative steps and design testing scenarios so that developers can detect errors at an early stage of the project and fix them quickly. Nevertheless, some CubeSat developers may still prefer to implement traditional methodologies, such as the v-model or waterfall models for their CubeSat projects. An example CubeSat project life cycle is depicted in Figure 1.4.

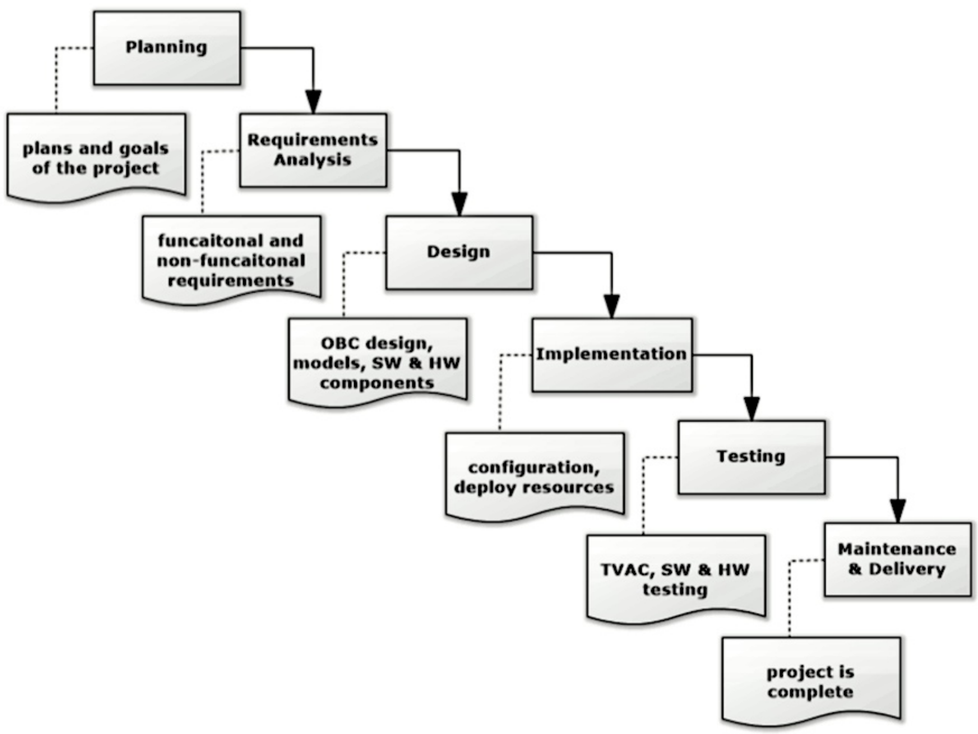


Figure 1.4. CubeSat project life-cycle (based on [25]).

The time required for each phase will vary. The concept phase, for instance, may take up to six months, whereas developing and testing the ground station may take a whole year [25]. Figure 1.5 shows a notional timeline for project phases. Some phases can, and typically do, run concurrently.

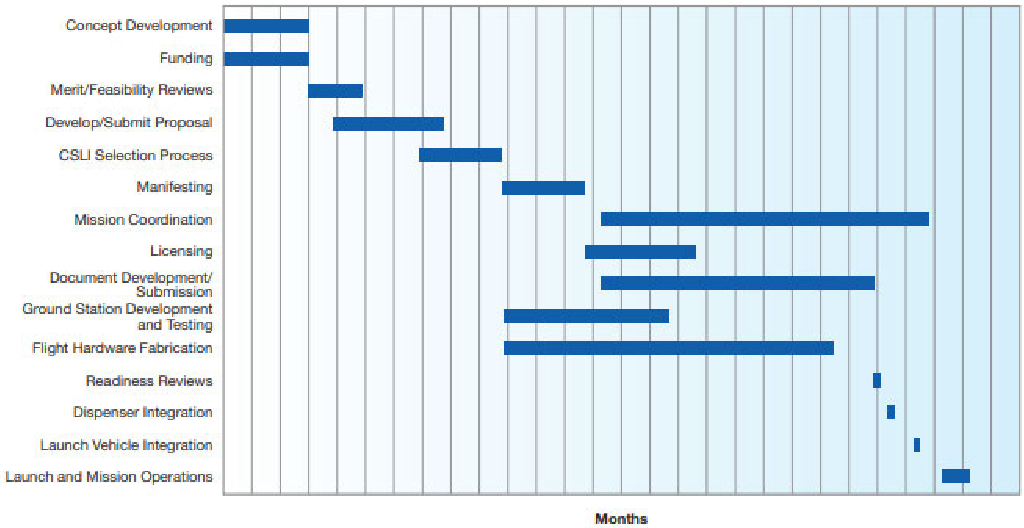


Figure 1.5. A timeline of a CubeSat project's phases [7].

1.5. Systems Engineering Principles and Methodologies

Systems engineering (SE) is a process that helps developers to successfully design and operate complex systems. The principles of systems engineering focus on effective ways for managing changes and complexity in systems [26]. SE can provide a better interpretation of system's requirements by allowing developers to understand stakeholder needs, and facilitating requirements elicitation and documentation [27]. SE processes and activities can be applied using various systematic approaches during the project lifecycle.

The project lifecycle initiates with phases that are associated with designing a comprehensive, iterative, and recursive problem-solving process that includes both management and technical components [27]. This is a systems engineering process that play a major role in defining logical sequence of tasks for a complex system. There are different methodologies and approaches that can be applied to a CubeSat project. For instance, the NASA project lifecycle, depicted in Figure 1.6, is widely used in developing cyber physical systems. It contains phases from pre-phase A to phase F that emphasize the main concept and mission objectives of the project, as well as the preliminary and critical design reviews, to insure the integration of subsystems components.

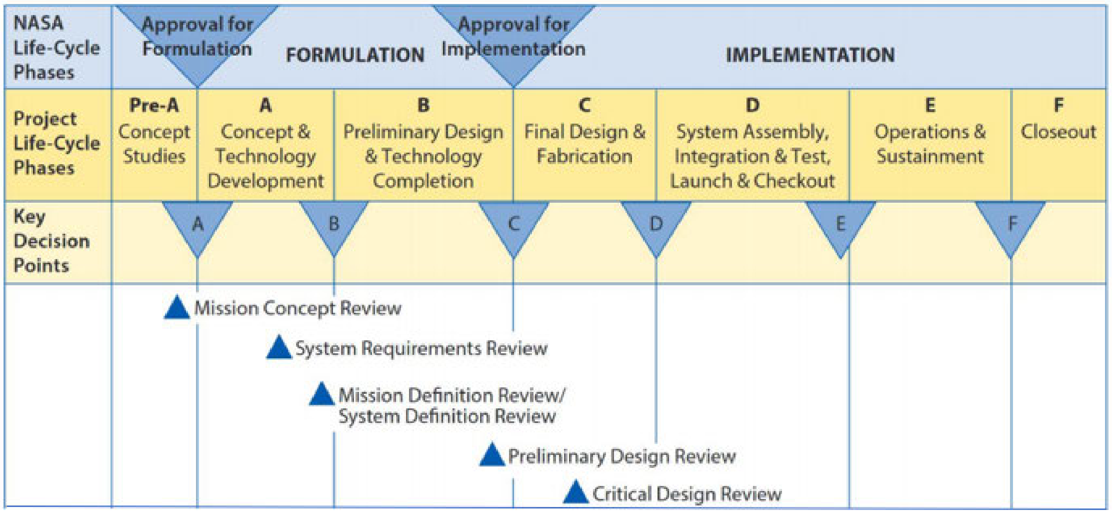


Figure 1.6. NASA project lifecycle [25].

Another common development lifecycle model is the V-Model [28]. The V-Model, illustrated in Figure 1.7, has multiple levels of development as a system evolves from the concept of operations and user requirement identification stage, through to detailed design and verification to final system validation.

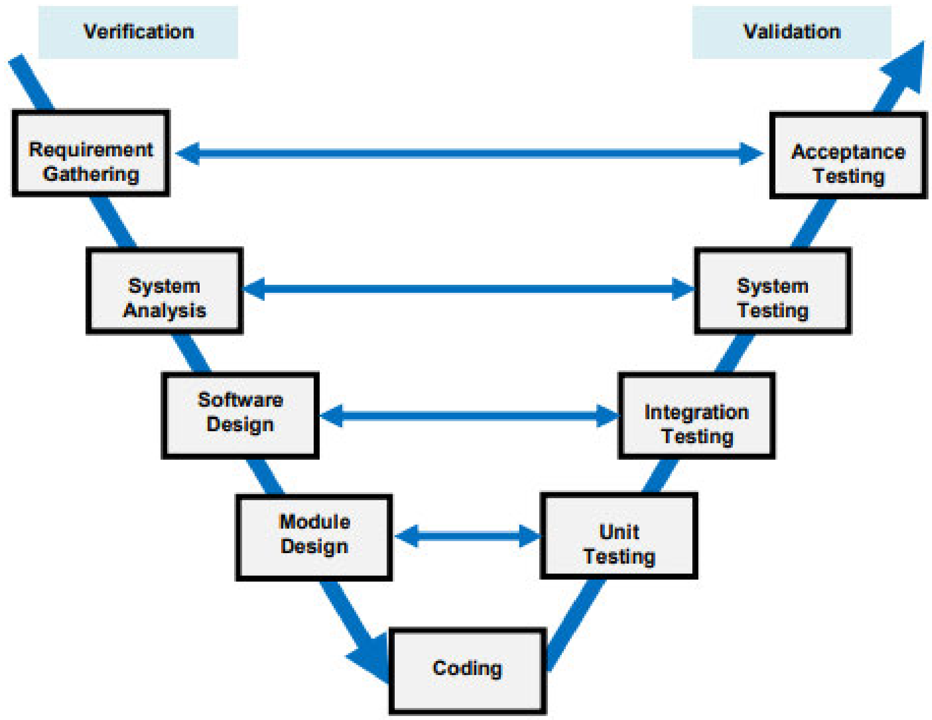


Figure 1.7. The V-Model [28].

The spiral model [29], depicted in Figure 1.8, is another common project lifecycle model. It consists of four phases of systems engineering activities: planning, risk analysis, engineering, and customer review. Requirements are divided into small iterations, called spirals, and developed and tested through the whole project. Although this model emphasizes risk analysis and might be beneficial for large projects, it is rarely discussed in the development of educational CubeSat projects.

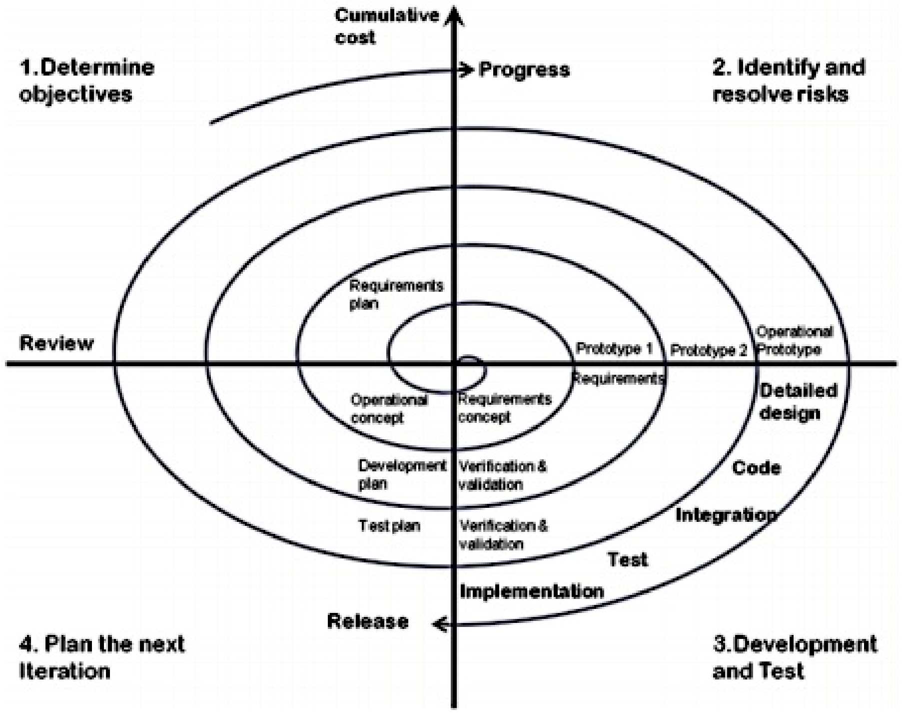


Figure 1.8. The Spiral Model [29].

The iteration model, which is also a common development lifecycle, is based on two assumptions. The first assumption is that implemented use cases in one iteration are independent from those in other use cases. The second assumption is that the amount of work necessary to re-factor existing designs and implementation to accommodate the new functionality is much less than to implement the new functionality [30]. This model is illustrated in Figure 1.9.

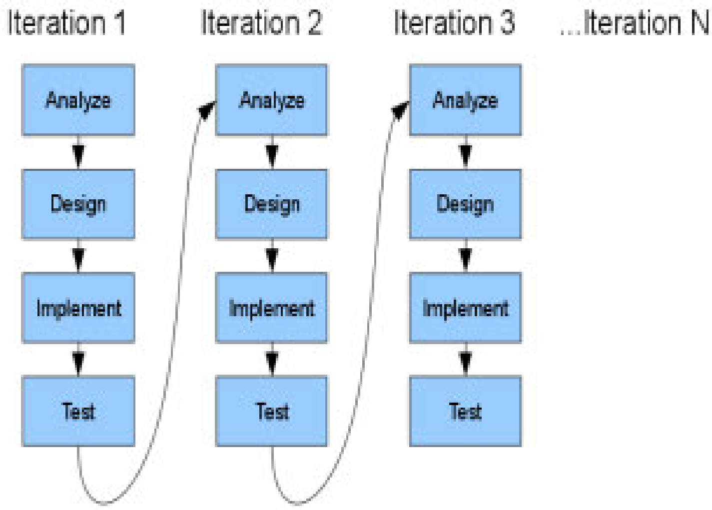


Figure 1.9. The Iteration Model [30].

1.6. The High Demand for CubeSats

Prior to the advent of CubeSats, the space industry primarily built expensive, large satellites that were developed through complex and rigid processes and procedures [31]. CubeSats were initially not attractive to industry professionals and space scientists, when they were first introduced. However, over time, they revolutionized the space industry and created an alternate to the complex process of spacecraft manufacturing with the introduction of a development environment with lower hardware and labor costs [32]. CubeSats have been proposed or used for different types of missions, including tracking space debris [33], detecting magnetic fields [34], Earth observation [35], and detecting gamma rays [36]. Using COTS components played a major role in CubeSat popularity because developers can easily build and test their CubeSat in a short development cycle.

Historically, building a space mission has been very complex and expensive. It would cost millions of dollars and take ten years or more of development [14]. In comparison with traditional satellites, CubeSats are in high demand for several reasons, including [14]:

- Flexibility with adding or removing components
- Using open-source software and hardware components
- Less expensive

- Can be developed in less than two years, in some cases

CubeSats have been proposed for interplanetary missions such as missions to Mars [36]. The future possibilities for CubeSats are increasing, as they are seen to be successful in a variety of types of missions. Additionally, CubeSats have been proposed with longer mission durations: as high as 25 years [3].

This high demand for CubeSats exists even despite a high rate of failure of CubeSat systems. This failure rate is due to several reasons, including the tools and models used, and students' limited level of experience [58]. Some researchers have suggested, depending on the CubeSat mission and the budget of the project, that universities and manufacturers should even mainly focus on the system and component level design, while neglecting the elicitation of requirements [58]. This is inheritantly problematic.

1.7. Dissertation Contents

This dissertation is presented in seven chapters. It continues with Chapter 2, which provides a literature review. In this section, CubeSat background and the reasons for CubeSat failure are discussed. This chapter also presents related studies regarding approaches to increase CubeSat mission success.

The third chapter presents the research methodology. This includes six factors associated with CubeSat mission success. It also discusses the research question and research hypotheses and the collection of data for this research. The fourth chapter discusses the survey design. It explains how survey questions were designed for this research. The fifth chapter includes the results of the statistical analysis for the data. The sixth chapter presents the results and explains why they are important. Finally, the seventh chapter includes a summary of the work presented in previous chapters and recommendations for CubeSat developers.

This research aims to answer a key research question: Is the adoption of a systems engineering methodology effective in reducing CubeSat mission failure? To answer this research question, this study investigates the reasons for failure and presents factors associated with CubeSat mission success. The research focuses on evaluating the benefits of applying systems engineering methodologies and how effective they can be in reducing CubeSat mission failure rate. It includes six hypotheses that are statistically tested to answer the research question.

2. LITERATURE REVIEW[2]

A variety of factors have led to frequent CubeSat failures. Tracking the reasons for these failures can help developers build more reliable and robust CubeSats. While several researchers track successful CubeSats [6][27][39] there are a limited number of researchers [58][62][88] that track failures issues and investigate the causes of these failures. This chapter considers the reasons for CubeSat failures and tools and tools and practices to support CubeSat development and drive future missions' success.

2.1. Background

When establishing their projects, CubeSat developers, such as students and hobbyists, rely heavily on open source components [66]. The Open Source Initiative (OSI), defines open source software as a software that can be freely used, changed, and shared by anyone. Many people make open source software and distribute it under licenses that comply with the forgoing definition [37]. The Open Source Hardware Association (OSHA) defines open source hardware as any hardware whose design is made publicly available so that anyone can study, modify, distribute, make and sell the design or hardware based on that design [38].

Open source products are beneficial to CubeSat developers because they support reliability, customization, collaboration, innovation, and cost [38]. This, however, increased the popularity of CubeSats in the last 10 years [38]. This high demand for CubeSats, is even despite a high rate of failure of CubeSat systems. This failure rate is due to several reasons, including tools and models used, and students' limited level of experience. Some researchers [39][40] have suggested, depending on the CubeSat mission and the budget of the project, that universities and manufacturers should even mainly focus on the system and component level design, while neglecting the elicitation of requirements. This is inheriting problematic.

Swartwout studied Small Satellites and presented data for success and failure for missions between 1994 and 2017 [5]. The data showed the growing use of CubeSat systems. It also showed that roughly 40% of CubeSats failed to meet their mission objectives. This work also discussed

[2]Based on: A. Alanazi, A. Jones and J. Straub. Requirements Modeling Language and Automated Testing for CubeSats. Proceedings of the 2019 IEEE Autotestcon Conference.

that the concept phase, where the CubeSat mission and goals are identified, may be neglected by some university programs, who are willing to accept higher risk of mission failure. The number of launched CubeSats is rapidly growing, exceeding 1200 launched CubeSats to orbit since 2003 [39]. The current number of CubeSats and SmallSats can be found in Swartwout and Kulu's databases [6].

2.2. Reasons for Mission Failure

The failure rate among CubeSat missions is high due to their fast orbit decay and reentry because they are launched into low-Earth orbit (LEO). Commonly, within two years, they decay to a reentry trajectory [40]. This limited window limits the time available for on-orbit troubleshooting. Another reason for failure is the practices used by some university-class CubeSat projects [1]. Some projects have low quality standards during the manufacturing, assembly, integration, and validation phases [41]. For successful mission assurance, systems engineering processes, such as quality control, are needed during the entire CubeSat development lifecycle. CubeSat missions also fail due to functional failures (i.e., not design and/or manufacturing-related failures) [41][42]. Another reason for failure is the lack of time and lack of resources, especially in the later stages of a CubeSat project [43][44].

In prior work, statistical analysis of factors associated with the failure of CubeSat missions was performed [45]. The listed factors were selected and identified based on feedback from an initial focus group of CubeSat developers:

- Testing time reduction (variable name: TTR)
- Design problems (variable name: DesPr)
- Availability of model for modification (variable name: Mod)
- Ease of addition or deletion of components (variable name: AddDel)
- System design objectives met (variable name: SysMet)
- Mission objectives met (variable name: MMet)
- Whether one model was employed as a reference model for different missions (variable name: Mission)

Data analysis regarding CubeSat failure was also included in Castet and Saleh's database. They collected data from 1584 satellite missions and discussed the reliability of subsystems. They also considered data analysis and modelling [46][47]. Other studies focus on the behavior of CubeSat subsystems using parameters, such as parametric analysis, to evaluate a range of values for an intervention [48].

Parametric and nonparametric analysis can help CubeSat developers to examine the behavior of subsystems. Parametric analysis distribution modelling was presented in [48]. The reliability of small satellite systems was also discussed by Guo, et.al., both parametric and nonparametric models were built and discussed [49].

2.3. Approaches to Increase CubeSat Mission Success

Some educational institutions have applied the concepts of guidance and specifications to help developers achieve mission success. For instance, the California Polytechnic University in San Luis Obispo, developed CubeSat specifications and made them available for use by the public [3]. Additionally, the NASA Jet Propulsion Laboratory CubeSat testbed included a summary of CubeSats development practices [31]. North Dakota universities have also developed components for an open source satellite [50][51][52]. The goal of this product was to provide reference designs and models to improve the output of educational CubeSat projects. These products can have tremendous learning benefits for students [53][54].

Many student educational programs for space studies are designed by space agencies to help engage students in STEM learning. Since 2013, Fly your Satellite, has been offered for students by the European Space Agency (ESA). This program is designed to involve college students in real-world space projects. It provides technical support, training, and funds some university-class CubeSat projects [55]. A group from CalPoly was among the participant groups with the FYS program. Their work included the analysis of Earth observation missions and Global Navigation Satellite System Reflectometry (GNSS-R) missions [56][57].

To attract students to space exploration and STEM education, NASA initiated a program called the Educational Launch of Nanosatellites (ELaNa). ELaNa is designed to involve university and high school students in the design and development of space missions, including assembling and testing payloads. This program is managed by the Launch Services Program (LSP) at NASA's Kennedy Space Center in Florida [58].

CubeSat developers commonly discuss their successful missions and share discussion with the public, but they rarely present the details of failed missions [59][60]. Because of this, most publications on CubeSat projects focus on mission descriptions rather than discussing the technical causes of CubeSat mission failure [61][62]. Missions' contributions to the engineering disciplines include brief discussions of the description of CubeSat missions [63], and guidelines for CubeSat developers, including designing and testing practices [59]. NASA has also published guidelines for new CubeSat developers, including students and hobbyists [7].

Other institutions have used quality assurance (QA) activities and practices to enhance project success. QA is defined as a set of measures applied to a CubeSat project to ensure consistency and to meet the expectations of stakeholders. Assurance activities should address materials, processes, procedures, and activities in the project lifecycle. ESA standards on QA are based on activities that largely occur during the design phase of a CubeSat mission [64]. Since documentation is one of the main factors in producing CubeSat mission success, ESA requires documentation for CubeSat projects [65]. However, excessive documentation could also lead to failure. Thus, ESA is trying to simplify their QA and documentation requirements for CubeSat missions to only the essential elements, after noticing a huge amount of redundancy in documentation requirements [23][66].

Weitz discussed other factors that lead CubeSat missions to fail [66]. This research explained that a lack of software requirements in CubeSat projects occurs due to a limited number of documented processes. Despite its benefits, Weitz identified some problems with the Consortium Requirements Engineering (CoRE) method, when applied to software requirements used by the PolySat program (the CalPoly student satellite program). Implementing the CoRE method can help in providing descriptions of the acceptable software behaviors. It shows relationships between variables derived from their requirements [67]. When applied, however, problems can occur with the level of requirements that the method generates. Problems occur when function-level requirements are combined with higher-level requirements. This sometimes forces developers to use certain components to insure the functionality of the spacecraft. However, this can also lead the development team to add more design elements to the requirements [68].

Compared to large aerospace projects, CubeSat projects are limited in team size, budget, and timeframe. Weitz explained that on average, 10 people will be on a CubeSat team [66]. Due to the small number of engineers, much less software engineers, working on a single project, the percent of time focused on design, implementation, and testing is increased while documentation time is typically reduced. This leaves requirements to be informally defined, if they are defined at all [68].

Straub, et.al., discussed that using the Requirements Modeling Language (RML) for a CubeSat project can enhance its validation and verification processes [69]. Applying RML in an engineering system development project (i.e., a CubeSat) will enhance the quality of requirements as well as easing the analysis and elicitation processes. RML is a language designed specifically to visually model requirements for easy understanding by stakeholders. RML provides management tools that help organize the phases of the entire project lifecycle [70]. Other models, such as UML, are commonly used in systems engineering development projects. While they might present a logical structure of requirements and features of an engineering system clearly to a developer or a tester, they can be hard to understand and complex when presented to stakeholders [71]. RML provides objective models that can visually present functional and nonfunctional requirements for better analysis and elicitation activities. RML also helps CubeSat developers to group and prioritize their requirements and to identify ambiguity or inconsistency easily.

RML uses a collection of diagrams to model software from the business analysis or product management perspective [69]. RML is concerned with a project's goals and objectives, instead of having complex system design models (which is the focus of UML and SySML). Moreover, RML models use boundaries based on different sections of the system, which are intended to bound to the problem space [72]. A requirements-mapping matrix (RMM), which is an objective model in RML, can be used by a requirement management model tool to automate the process of checking for missing links [70]. This is intended to ease the process of stakeholders understanding and analyzing the requirements. RMM contains multiple levels of mapping that maps process flow steps to requirements, requirements to business objectives, business objectives to features, features to requirements, requirements to code, and requirements to test cases. Using cross-functional charts which can illustrate how a process flows across organizational boundaries. This can help in identifying delays, redundancy, excessive inspection, rework and potential points of process failure [70]. RMM supports the fundamental limitation of human brain theory proposed by [70], which stated that a human brain can only remember seven plus or minus two tasks at once. Thus, RMM only includes eight or less columns in its matrix. The first three columns (L1, L2, L3) are process flows, and the rest of the matrix includes business objectives, requirements, business rules, code, and test cases.

Other studies focused on the importance of using tools and methods to increase CubeSat missions' success rates. Emami, et.al, [73] presented results from a project conducted at the Lulea University of Technology in Sweden and provided technical details for phases from design to operations for a 2U CubeSat. The report also includes details about the development processes used and the verification and operations of the CubeSat. It also explains the different issues that exist in CubeSat development, covering a structured lifecycle for a CubeSat project. These include software testing procedures such as test-driven development (TDD), testing time, quality assurance and documentation. This study pinpointed issues with time availability that can lead a CubeSat project to be delayed or fail. It suggests the need to include a longer time for functional tests, whenever a new software version is introduced. Also, to maximize communications time, building a single ground station for the satellite is not sufficient. Thus, it suggests that additional time should be budgeted to coordinate with other institutions in different locations to partner up on communications for the project [74][73].

Reza, et. al. [71], discussed how universities can apply different engineering models and tools for developing CubeSat systems. For instance, the University of North Dakota used the model-based systems engineering methodology (MBSE) to simplify and expedite requirements elicitation. They designed quality-attribute (QA) scenarios to document the non-functional requirements (NFR) of the CubeSat. Once a MBSE project has been created, specific attributes scenarios can be added. Each of those attributes must fall under a specific category. These categories are quality attributes decided upon by the project lead (such as availability and maintainability) [71].

Reza, et. al., also discussed methods that support the selection of systems architecture elements using nonfunctional requirements (NFRs) [75]. This paper discussed tools that support the automation of requirements processes such as Visure quality analyzer and inteGREAT [76]. Visure quality analyzer can assist in quality assessment and improvement. It allows the developer to elicit, define, asses, improve and manage the quality of individual requirements and complete requirement specifications [75][77]. The tool inteGREAT helps in use case modeling and can enhance the traceability of requirements [78]. The paper also discussed tools developed by Bright Green Projects, Leap SE and SPACE [75][79].

Straub, et. al., considered technical challenges with the requirements specification of CubeSats [80]. This paper covered the fundamental challenges of NFRs for CubeSat systems including security, usability, safety, deployability, configurability, constraints of size and weight, and overall performance. It discussed the possibility of using and modifying several software and systems engineering models and products including Model-Driven Engineering [80][50] , Model-Based Engineering [81], and the Architectural Analysis and Description Language (AADL) [82][83][84].

MBSE models have been used previously for space systems design and development. They can enhance the documentation processes, which especially is pertinent to the identified CubeSat development challenges. Selic studied issues related to requirement documentation for cyber physical systems (such as CubeSats) [85]. De Niz discussed embedded UML, which allows for simulation and formal and informal analysis of a model [86]. He compares AADL with UML to identify the areas that should be utilized during the software development activity [86]. Jürjens introduced UMLsec, which is an extension of UML that focuses on modeling design for secure systems [87][88].

Straub and Whalen presented a space program operating at the University of North Dakota, named the Open Prototype for Educational NanoSats (OPEN) [50]. This program includes traditional STEM activities (e.g., spacecraft engineering and software development); it also incorporates students from non-STEM disciplines that are not generally involved in aerospace engineering projects such as management, entrepreneurship, education, and fine arts. The objectives of this program are to increase students' proficiency in technical areas, such as spacecraft design and development, and to provide students with leadership opportunities in numerous areas and enhance their soft skills [51][53].

Weisgerber, et. al., presented a case study on a 1U CubeSat designed at the Technical University of Munich [88]. A team of students designed 3D printed prototypes for better representations of their CubeSat and the recruitment of new team members. The team also implemented a 3D printed CubeSat, at an early stage, to help students validate and verify CAD models to avoid potential design deficiencies [88]. This group proposed that an estimation of functional testing time for subsystems, at an early stage of the CubeSat development lifecycle, would increase the reliability of the CubeSat. A tool was developed, based on Bayesian methods, to help CubeSat developers determined the required functional testing time and reliability needs for their systems [49].

The results of the analysis of system and design risks for small satellites are discussed in [49][42][89]. Another case study, presented by Gonzalez, et. al. [89], at the Universidad del Valle de Guatemala, discussed different methodologies that can help CubeSat developers to design their mission using risk analysis. The team at the Universidad del Valle de Guatemala investigated the risk factors that might lead to a CubeSat mission's failure. They designed a tool (risk matrix) to mitigate risk factors during the design phase of their project [90]. Their tool works as a filter, when there is a high-risk technical challenge that exists, so that they can accommodate this when designing their CubeSat system. This matrix was based on a scoring model. The approach is simple for projects where it is easy to add to and edit the scoring system. The model can be optimized, so different organizations can use it when developing their CubeSat mission by simply providing their values for each parameter.

Kaslow, et. al. [90], present a system engineering model developed by the Space Systems Working Group (SSWG). The SSWG is a group of individuals working on small satellites with interests in systems engineering. The group presented their model developed using the systems modelling language (SysML), that helps developers enhance their diagrams with descriptions of requirements, parametric values, and structures. Their model provides a logical architectural structure for CubeSat development. Since a lack of documentations is considered one of the key failure reasons for CubeSat systems, the SSWG group contends that their model will work better than traditional "document-centric" systems engineering. SSWG models are created on an as-needed basis. Generating requirements comes from analyzing different aspects of a CubeSat mission's system performance requirements [81].

SSWG discussed several engineering processes that show the ability to model behaviors, interface with COTS simulation tools, and carry out trade studies. Currently, the team is building a reference CubeSat model for CubeSat developers. The model includes minimum needed specifications and can be applied to different CubeSat missions [91],[92],[93].

Gagliardi, et. al., [24] discuss how failing to address quality attributes at early stages in a system's architectural design can lead to operational and developmental failure. Quality attributes failures in areas including availability, reliability, usability, and maintainability can have a high cost, if they are not addressed at an early stage of design. This paper discussed the mission thread workshop (MTW), a stakeholder-centric workshop. MTW is used to elicit requirements and refine end-to-end quality attributes. Stakeholders can engage in the system vision process in the early life of a project by adding their inputs and prioritizing the importance of quality attributes. This tool can also be used to enhance the team's understanding of the architectural design of the system [24].

3. RESEARCH METHODOLOGY[3]

The goal of this research is to develop an engineering methodology that includes a set of processes to help reduce failure rates and increase success rates for future CubeSat missions. Thus, this research will work towards answering the following research question (RQ): is the adoption of a systems engineering methodology effective in reducing the rate of CubeSat mission failure?

This study uses a hybrid method of research. It utilizes a mixture of both qualitative and quantitative types of questions. Both methods are applied to investigate the relationships between factors that are associated with CubeSat mission success. The qualitative method provides insights into and understanding of the reasons for the frequent failure of CubeSat missions, while the quantitative method quantifies the collected sample data and generalizes its results.

3.1. Associated Factors of Mission Success

This study evaluates six independent variables (factors) that are associated with CubeSat mission success. It explains how these factors correlate with mission success, and how they are used in developing the final model. Each of the factors has been identified and coded for statistical tests.

3.1.1. Defining Mission Objectives (DMO)

Defining the mission scope, project scope, and success criteria is a critical and important factor in ensuring mission success. Similarly, outlining mission objectives is key to meeting strategic CubeSat mission standards. It also helps in understanding how mission objectives are met, and what factors impact mission objectives. However, because CubeSat developers' perspectives vary by area of technical focus, answers may differ from one organization to another. Some developers might indicate that their CubeSat project did not meet some or all their mission objectives, but they may still consider it a success.

Success criteria may also vary by mission type. For example, the definition of mission success for students may be completing their project within time and budget targets, whereas for

[3] Based on A. Alanazi and J. Straub. 2020. Defining CubeSat Mission Success and Assessing Reasons for Mission Failure. Under preparation for submission to the International Journal of Aerospace Engineering. and A. Alanazi and J. Straub. 2020. Evaluation of Software Engineering Practices Application to CubeSat System Development. Under preparation for submission to Acta Astronautica.

industry professionals, a successful mission would be expected to survive and successfully operate for a 60-day or longer period on orbit. Other organizations may consider their CubeSat mission a success by just establishing a communications link [94]. The hypothesis statement (H_1) derived from this factor is: clearly defining mission objectives (DMO) at an early stage of a CubeSat project's life cycle has a positive effect on the mission's success rate.

CubeSat development teams may never discuss the scope and objectives of their mission; however, that does not always lead to failure of the mission. Nevertheless, this study is attempting to measure the number of CubeSat missions that have succeeded, and the number that clearly defined their mission at an early stage in the project's life cycle. Additionally, the study will focus on the investigation of how CubeSat missions have succeeded without defining their mission objectives.

3.1.2. Timeline Analysis (TA)

It is generally considered essential to define a detailed testing plan, including an amount of time to be spent, for each stage of the project life cycle. It is equally important to know what methods of testing were performed on a CubeSat system, and to know the amount of time that was allocated for each test [95]. Testing time may vary due to the complexity of the mission. It is also crucial, for planning future missions, to determine if the CubeSat's design helped in reducing the testing time required. Thus, adding a timeline analysis technique, which is a critical systems engineering process, to a CubeSat project will clarify the relationships between functions and tasks, and will identify the specific time allocated for design requirements.

Timeline analysis can show the required time needed, as well as the design constraints in place, while building CubeSat systems. TA can help in defining how much time is needed for each task; thus, it will help developers in dividing up the work to complete their tasks. This will also be beneficial in prioritizing tasks and deciding which ones are assigned longer time periods as compared to others [96]. Time is a crucial factor in CubeSat systems development because hardware and software systems are being tested within a timeframe that may be constrained by launch providers and other deadlines. Thus, a campaign needs to be designed that defines the type of tests, designs fixtures, and identifies laboratory facilities that can perform the tests according to the documented specifications. The hypothesis statement (H_2) derived from this factor is: implementing timeline analysis (TA) in a CubeSat project has a positive effect on the success rate of CubeSat missions.

This research investigates the association between the use of timeline analysis and the success or failure of a CubeSat mission. It evaluates whether running timeline analysis for a CubeSat project is critical to success. It is used to identify the time related to specific design requirements. It also can help identify overlapped relationships between functions and tasks. It was expected that a higher number of professional CubeSat developers would have used or discussed TA, as compared to developers in educational institutions. One reason for this is that this type of analysis requires more time, a bigger budget, and a larger team to be performed. It also requires experience in forecasting the required time for testing.

The study also investigates if (and how) CubeSat developers were able to reduce the time needed for system testing, without impacting their CubeSat mission's success. This will help CubeSat developers in designing a timeline analysis platform and enable them to use the minimum amount of time needed for achieving sufficient system testing quality.

3.1.3. Mission Assurance (MA)

The mission assurance factor is closely related to the DMO factor. Mission assurance depends on mission risk management, which requires missions to be well defined. Running a risk assessment test is very important in ensuring a mission's success. This test will help in identifying flaws and issues at the early stages of the CubeSat development. It is also important to know if developers have used or ever discussed risk-based practices. The data was expected to include an indication of whether one or more mission assurance tests has been performed by a developer group.

Mission assurance tests are essential to a CubeSat mission's success. Tests may include, but are not limited to, communication link testing, power system testing, thermal & vacuum testing (TVAC), and regular reviews (formal and informal). The hypothesis statement (H_3) derived from this factor is: performing mission-assurance analysis (MA) in the CubeSat project lifecycle has a positive effect on the mission's success rate.

The study investigates if using an MA process will lead CubeSat missions to succeed. The other consideration is how CubeSat projects that never performed MA also end up succeeding. These results will be combined to see if MA plays a major role in mission success. If correlation between successful CubeSat missions and including MA in their projects is shown, CubeSat developers would be well advised to add MA to their development model using a quality risk management tool and constraints, if needed.

3.1.4. Critical Design Review (CDR)

Complexity of system design is very common in CubeSat development. However, a simple design could be critical to a mission's success in some environments. Survey respondents were asked to identify the best practices, to describe their experience with design problems and issues, and if they could provide an alternate approach to avoid future mission failure.

A critical design review (CDR) is performed during the system development phase. It may include a series of reviews conducted for hardware and software systems, coding, and testing. Additionally, test plans are reviewed to assess if test efforts are developing sufficiently. The approved detailed design serves as the basis for final production planning and initiates the development of final software code. The CDR may also include metrics for measuring design optimizations and constraints. The hypothesis statement (H_4) derived from this factor is: performing a critical design review (CDR), in the design phase, has a positive effect on the mission's success rate.

This study investigates how CDR is associated with CubeSat mission success. Developers with more experience and a greater team size were projected to include CDR in their documentation to enhance reusability, testability, and modifiability. Also considered is developers that only utilize a preliminary design review (PDR), and the success or failure of their mission. Results will be compared to assess the association between the design review used and the degree of mission success. This data will also be analyzed in conjunction with data collected from the DMO variable. The results will then be used to help CubeSat developers in developing models and in deciding whether both PDR and CDR need to be utilized, or if CDR alone will be sufficient to ensure a successful CubeSat mission.

3.1.5. Experience in Quality Attributes (EQA)

To decrease the failure rate and increase the success rate of CubeSat systems, they need to be secure, modular, and reusable. Thus, it is essential to study the challenges that have been experienced by CubeSat developers. Lessons learned by developers help them to succeed in their future missions. Analysis of these factors can identify lessons learned from one mission to share with others.

This factor will measure developers' experience with quality attribute practices including usability, security, and reliability. It will also measure the correlation between mission success and having expert personnel on the team, specifically, respondents will be asked about the number of years they have spent in this environment, and the number of CubeSat projects they have been involved with. The hypothesis statement (H_5) derived from this factor is: experience in quality attribute (EQA) practices has a positive effect on the mission's success rate.

This study investigates to what extent CubeSat developers who have experience in quality assurance practices have higher success rates in their CubeSat missions than other developers who do not. If, for instance, a higher number of successful missions include one or more EQA experts in the team, then a CubeSat developer should use a model that includes quality attribute practices. These quality attributes can be divided into three or more categories, including operation, development, and sustainment. The operational category will focus on the availability, usability, and overall performance of the system. The development category will focus on the modifiability and testability of the system, and sustainment will focus on the maintainability and deployability of the CubeSat system.

3.1.6. Schematic Diagram (SD)

The failure of components is closely related to overall CubeSat mission failure. Many believe that to have a robust CubeSat system, developers must use commercial-off-the-shelf (COTS) parts that are hardened and can survive in the space environment [96]. Thus, it is essential to know where failures occur, and why. Expected data included responses regarding deployment testing, the compatibility of radio frequencies (RF), and issues with the power system.

The schematic diagram (SD) is a systems engineering process that can depict hardware and software components and their interrelationships. It also can include design standards for avionics. SDs are developed at successively lower levels as the analysis proceeds to define lower-level functions within higher-level requirements. These requirements are further divided and allocated using a requirements matrix (RM). The hypothesis statement (H_6) derived from this factor is: including a schematic diagram (SD) in a CubeSat project has a positive effect on the mission's success rate.

Including a SD in a CubeSat project is expected to help ensure that a robust CubeSat system results. However, flaws in a SD may lead to overall mission failure. Thus, this study investigates to what extent CubeSat missions fail due to models and diagrams and to what extent tools cause or contribute to failure. Knowing where defects in models exist and how to fix them is also important. Expected answers from CubeSat developers may vary due to their type of mission and how complex their system is. Data collected from these factors will help CubeSat developers to design a high-quality SD connected to a matrix of requirements.

Including a SD in a CubeSat project will provide visibility of related system components and traceability to the requirements matrix and other system engineering documentation. SDs document the solution to the functional and performance requirements established by the functional architecture. They show interfaces in between the system components and other systems or subsystems. They also support traceability between components and their functional origin and provide a valuable tool to enhance configuration control.

3.2. Research Hypotheses

The research hypotheses are specific, testable, and designed to provide initial answers to the research question. They state predicted relationships between independent and dependent variables. The following hypotheses statements have been constructed based on the list of associated factors discussed in Section 3.1.

H_1: Clearly defining mission objectives (DMO) at an early stage of a CubeSat project-lifecycle has a positive effect on the mission's success rate.

H_2: Implementing a timeline analysis (TA) in a CubeSat project has a positive effect on the mission's success rate.

H_3: Performing a mission-assurance analysis (MA) in a CubeSat project lifecycle has a positive effect on the mission's success rate.

H_4: Performing a critical design review (CDR) in the design phase has a positive effect on the mission's success rate.

H_5: Experience in quality attribute (EQA) practices has a positive effect on the mission's success rate.

H_6: Including a schematic diagram (SD) in a CubeSat project has a positive effect on the mission's success rate.

Each of the hypotheses will be statistically tested to measure the association between the independent and dependent variables. A null hypothesis (H_0) is also needed, which is a default position where there is no association between variables. The null hypothesis is presented in the following statement:

H_0: Each independent variable has no effect on the CubeSat mission's success rate.

3.3. Data Collection

A survey was used to investigate the reasons for CubeSat mission failure and to identify tools that can increase a CubeSat mission's success. The survey has been designed to measure the challenges and needs of CubeSat developers and to investigate and identify the best engineering practices and techniques that are associated with successful CubeSat missions. The survey was distributed to CubeSat systems developers. This includes students, faculty, researchers, and industry professionals who are associated with CubeSat development projects for scientific, educational, and commercial purposes. The target sample size of this study was 200.

4. DESCRIPTION OF SURVEY[4]

A survey was designed for administration to CubeSat developers. The survey included questions related to multiple factors prospectively associated with CubeSat mission success or failure. These included the possibility of adding or deleting components to/from the system design and system modifications' feasibility.

4.1. Survey Design

The survey consists of 32 questions. Five questions are open-ended questions for collecting qualitative data. A validity test (internal and external) was conducted after designing the survey questions. This test illustrated how the findings are presented and helped indicate how accurate the survey questions are. The data collected from the open-ended questions was analyzed and coded using NVivo12 to analyze the textual data. Data is stored as nodes (cases). All of responses to a certain question are grouped together under a single node. NVivo12 includes auto-coding techniques that can automatically identify themes and sentiment in the responses to open-ended questions. The coded data then was aggregated with the rest of the data and was statistically analyzed using SAS.

For the analysis process, different statistical techniques were used including linear regression, the Chi squared test of independence, logistic regression, and the Kruskal-Wallis test for independence to determine the accuracy of the expected results. The statistical analysis of the survey included descriptive analysis, frequency analysis, and statistical tests. The statistical tests used for this research are the Chi squared test for independence and logistic regression. Logistic regression consists of a T-test, used for each independent variable, and a F-test, to test the research model.

The research model includes dependent variables (DVs), independent variables (IVs), and coefficients. Independent variables are factors that are associated with a CubeSat mission status, which include:

[4]Based on A. Alanazi and J. Straub. 2020. Defining CubeSat Mission Success and Assessing Reasons for Mission Failure. Under preparation for submission to the International Journal of Aerospace Engineering. and A. Alanazi and J. Straub. 2020. Evaluation of Software Engineering Practices Application to CubeSat System Development. Under preparation for submission to Acta Astronautica.

- Defining Mission Objectives (DMO)
- Timeline Analysis (TA)
- Mission Assurance (MA)
- System Design (SD)
- Experience in Quality Assurance (EQA)
- Schematic Diagram (SD)

Each factor was treated as an individual variable and statistically analyzed to test the association with CubeSat mission's success or failure. Therefore, to ascertain whether each one of the IV's is related to another, Chi-squared tests of independence are used with mission status as the DV and one IV for each test of independence. In addition, it is possible that two or more of the IV's can jointly explain the success or failure of the mission. For this, a logistic or binomial regression model was constructed with mission status as the DV and all the IV's. The dependent variable (DV) for this study will be "mission status", which has two values:

- Mission failure = 0
- Mission success = 1

4.2. Survey Validation

To validate the survey questions, two methods were applied: face validity analysis and Cronbach's alpha analysis. To assess face validity, experts from NASA's JPL and the Space Systems Working Group (SSWG), who have expertise in CubeSat development, carefully reviewed all of the survey questions and evaluated whether these questions accurately captured the topic under investigation. Additionally, a statistician reviewed the survey questions for common errors such as leading questions, double barreled questions and confusion.

Cronbach's alpha was applied as part the validation process to check the internal consistency of questions regarding these same factors. Cronbach's alpha is computed by correlating the score for each scale item with the total score for each observation, and then comparing that to the variance for all individual item scores [97]. Cronbach's alpha is the most common estimation tool to measure the internal consistency of items in a scale [98]. For this study, alpha is applied to measure the correlation between survey questions. It estimates the proportion of variance that is systematic or consistent in a set of survey responses. The Cronbach's alpha formula for computing α is expressed as follows [98]:

$$\alpha = \frac{k}{k-1}\left(1 - \frac{\sum_{i=1}^{k} a_{Yi}^2}{\sigma_X^2}\right) \tag{4.1}$$

where k represents the number of survey items in the scale and σ_X^2 is the variance of the observed total scores. Also, a_{Yi}^2 is the variance of item (i) for a respondent (Y). The size of α is determined by the number of items in the scale and the mean inter-item correlations. The acceptable range of Cronbach's alpha, $0.8 > \alpha \geqslant 0.7$, indicates good internal consistency of the items in the scale [99]. Table 4.1 shows the Cronbach's alpha values for questions in this study is 0.78, indicating good internal consistency.

Table 4.1. Cronbach Coefficient Alpha.

Variables	Alpha
Raw	0.7542
Standardized	0.7818

4.3. Survey Questions

The survey includes a set of questions about the experience of CubeSat developers, including their role in the CubeSat project, the number of CubeSats they have developed, their experience with QA, the challenges they have experienced during the design phase of the project, their perception of the development phase, and the status of their CubeSat mission. There are five open-ended essay questions. Responses to these were analyzed and coded using thematic analysis. A full survey instrument is included in the Appendix.

4.3.1. Open-Ended Questions

Five open-ended questions were included in the survey. Each is now presented.

How do you define a successful CubeSat mission?

Respondents were able to give more than one answer to this question. Therefore, thematic analysis is used to identify the common factors identified by the respondents. NVivo12 was used to help identify these themes automatically in all questionnaires. The findings of this analysis were used to identify the different perspectives of CubeSat developers regarding CubeSat mission success. Expected responses to defining a successful CubeSat mission included:

- If it was developed within time and budget.
- If it was completed on time and ready for launching.
- If it met the minimum mission objectives.
- If it was launched successfully.
- If it established a communication link with the GS.
- If it is sent data to the GS for two months or more.

What are the most challenging activities in a CubeSat development lifecycle?

It is probable that the perception of complexity depends on the experience of the project team. In addition, respondents might be involved with different activities in the CubeSat development. Potential data may include challenging activities in hardware, software, documentation, delayed schedules, and other issues. The data was analyzed using frequency distribution analysis. Expected responses to this question included challenges that exists in:

- Hardware or software components
- Lack of documentation
- Delayed schedule
- Shortage of COTS components
- Following methodologies and models

Do most of system designs fail due to tools or models? and why?

The collected data was expected to include whether designs typically fail because of tools rather than models, or if there are more failures due to models than tools. Some answers may include failures that are attributable to both tools and models. The expected data will help to identify the reasons for the failure of a specific area of CubeSat development. The data will be analyzed using frequency distribution analysis between tools and models. The types of failure with the highest number of responses were identified for subsequent analysis.

Has your system design reduced testing efforts and/or improved performance?

This question expected responses of yes and no. It was also expected that respondents would discuss the ways design has been beneficial and best practices that can help in improvement. Respondents were also expected to identify steps or processes to avoid in the design phase. Analysis of this question used frequency distribution information for the type of CubeSat mission, the number of respondents, and the efforts in design and testing. A Chi squared test was used to determine if there was a difference in the views of these professionals regarding the effectiveness of system design. For further analysis, collected data was also tested for significance based on the type of methodology used by the respondents.

If you are given a second chance to redesign your CubeSat, what would you do instead? and why?

The answers to this question were expected to depend on the developers' experience and the status of their mission. Respondents were expected to focus on certain areas and stages of CubeSat projects including, but not limited to, planning, requirement analysis, design, testing, operations, and maintenance. The collected data was analyzed using frequency distribution analysis. Responses for this section were also analyzed using frequency tables. The frequencies were used to calculate inferential statistics. Responses were also used to describe the various categories of data in the sample and, as such, have no measures of central tendency or dispersion.

4.3.2. Multiple-Choice Questions

Thirteen multiple-choice questions were included on the survey. Each is now discussed.

What is/was your CubeSat mission?

Responses to this question were analyzed using a frequency table, which included the type of a CubeSat mission, and the type of CubeSat. For each mission type, the frequency analysis of responses included mission types compared with CubeSat types. Collected data was used to develop a histogram.

What is/was the team size of your CubeSat project?

Responses were analyzed using a frequency table that considered the team size, the number of respondents and how many CubeSats they have developed.

What is your role in the CubeSat project?

Responses to this question included the role of the CubeSat developer in the project. The responses were analyzed to compare the indicated role with how many CubeSat project has the respondent participated in.

How many CubeSat projects have you participated in?

This question collects data regarding the level of experience of the CubeSat developer. The more projects the developer has participated in, the more experience he/she has gained from lessons learned. A frequency table was used to link the number of projects and the number of respondents.

The CubeSat project you are associated with is/was under development, incomplete, failed, succeed.

This question seeks identify the status of the CubeSat project that the developer has participated in. Collected data was cross tabulated with other factors such as mission assurance and level of experience.

The most associated factor of a CubeSat project success is:

This question presumes that respondents have a technical understanding of the project. Both descriptive and inferential statistics were used to analyze the responses. Choices included were: HW/SW sufficient testing, COTS HW/SW functionality, and requirements analysis and documentation. Results were tabulated using frequency tables.

The most associated factor of a CubeSat project failure is:

This question asks about the most associated factor with CubeSat project failure. The choices included were: HW/SW insufficient testing, COTS HW/SW failure, and lack of requirements analysis and documentation. Results were tabulated using frequency tables.

The CubeSat you are associated with is:

In this question, the respondent is asked about the type of a CubeSat that he/she is associated with. CubeSat types listed were 1U, 2U, 3U, and other. Responses were tabulated using frequency tables.

What is the approximate time needed to develop a typical CubeSat?

In this question, the respondent is asked about the approximate time needed to develop a typical CubeSat. Answers for this question included: 2 years or less, 3-4 years, 5-6 years, and 7 years or more. The approximate time needed could be dependent on the role of developer and the type of CubeSat missions they have been involved with. The responses were tested using analysis of variance (ANOVA) at a 5% level of significance. For each table, the differences within and between groups were noted as potential significant factors that determine the time needed to develop the project.

The most prominent aspect of software quality that is associated with a CubeSat mission success is:

Collected data from this question depends heavily on the developer's experience. Aspects of software quality play a major role in framing the engineering methodology for the CubeSat system. Data was cross tabulated with the role of the developer, and the type of the CubeSat mission.

The least prominent aspect of software quality that is associated with a CubeSat mission success.

Collected data from this question depends heavily on the developer's experience. Aspects of software quality play a major role in framing the engineering methodology for the CubeSat system. Identifying the least quality-associated aspect will help CubeSat developers to develop alternative quality scenarios for their desired CubeSat methodology. Collected data was cross tabulated with the role of the developer, and the type of the CubeSat mission.

Which of the following methodologies is most suitable for your CubeSat project? and Why?

This question asks about which methodology is primarily used for the respondent's CubeSat projects. Answers will depend on the developer's perspective, level of experience, and the types of missions the developer worked on. Answers included were: waterfall, V-Model, SCRUM, and other. The responses were cross-tabulated with the type of mission.

Which technique is best used for requirements elicitation?

In this question, the developer is asked about the technique and practices used for requirements elicitation. Common practices are included as answers, including: interviews, focus groups, prototyping, and other. The responses to this question were analyzed using frequency distribution.

4.3.3. Likert-Scale Statements

For each of the statements below, respondents were asked to choose the best response that characterizes the statement where, 1 was strongly disagree, 2 was disagree, 3 was neither agree nor disagree, 4 was agree, and 5 was strongly agree.

- CubeSat mission fail due to failure of tools rather than models.
- CubeSat mission fail due to failure of models rather than tools.
- Project scope and goals were clearly identified at the early stage of development.
- A methodology was carefully selected for a CubeSat project.
- Stakeholders can add/delete requirements in the CubeSat testing phase.
- Stakeholders participated heavily in the CubeSat design phase.
- A project team size has a positive effect on the CubeSat mission.
- Requirements were complex and hard to follow.
- Requirements were elicited and well documented.
- Requirements specification contains several non-testable functional requirements.
- The application of methodology has a positive effect on a CubeSat mission.
- Applied methodology enhanced the communication & control subsystems.
- A CubeSat project had attended to its original scope.
- A CubeSat project schedule was well managed.

The results from this section were analyzed using frequency tables. The average of the Likert scores was taken as the representative score for each question. The answers with the highest sum of Likert scores were graphed to illustrate the areas that need more attention in the CubeSat development life cycle. The analysis for this section used one-way ANOVA. However, when the data was non-normally distributed, the Kruskal-Wallis test was used to identify whether there were differences between the groups. This test indicated significance and established the presence of differences between groups.

This section has described the survey and its design. It has also discussed the expected data and outcomes from the survey questions.

5. RESULTS AND ANALYSIS[5]

A survey was distributed electronically through the Qualtrics survey platform to students, professors, researchers, and industry professionals who collaborated on CubeSat development projects for scientific, educational, and commercial purposes. Out of the distributed survey links, 127 were received and statistically analyzed.

In this section, the applicable results are discussed. These include the test for model effects for each hypothesis. Each associated factor of mission success (DMO, TA, MA, CDR, EQA, and SD) was treated as a model with its factors (CS_#). Each successful factor was developed based on more than one survey question. For the purposes of brevity, during the analysis process, each question was coded as CubeSat_# (i.e., define successful mission was CS_1). Table 5.1 lists the survey questions (numbers 1 to 18) and the Likert-scale statements (19 to 32), coded as CS_1 to CS_32.

[5]Based on A. Alanazi and J. Straub. 2020. Defining CubeSat Mission Success and Assessing Reasons for Mission Failure. Under preparation for submission to the International Journal of Aerospace Engineering. and A. Alanazi and J. Straub. 2020. Evaluation of Software Engineering Practices Application to CubeSat System Development. Under preparation for submission to Acta Astronautica.

Table 5.1. Survey questions with associated codes.

No	Question	Code
1	How do you define a successful CubeSat mission?	CS_1
2	What are the most challenging activities in a CubeSat development life cycle?	CS_2
3	What is/was your CubeSat mission? [Scientific, Educational, Commercial, Other]	CS_3
4	What is/was the team size of your CubeSat project? [3-6, 7-10, 11-15, 16 or more]	CS_4
5	What is your role in the CubeSat project? [Principle Investigator (PI), Systems Analyst, Software Engineer, Other]	CS_5
6	How many CubeSat projects have you participated in? [1-3, 4-7, 8 or more]	CS_6
7	The CubeSat project you are associated with is/was: [Underdevelopment, Incomplete, Failed, Succeed].	CS_7
8	The most associated factor of a CubeSat project success is: [HW/SW sufficient testing, COTS HW/SW functionality, Requirements analysis & documentation, other].	CS_8
9	Do most of system designs fail due to tools or models? and Why?	CS_9
10	Has your system design reduced testing efforts and/or improved performance?	CS_10
11	The most associated factor of a CubeSat project failure is: [HW/SW insufficient testing, COTS HW/SW failure, lack of requirements analysis and documentation, other].	CS_11
12	The CubeSat you are associated with is: [1U, 2U, 3U, Other].	CS_12
13	What is the approximate time needed to develop a typical CubeSat? [2 years or less, 3-4, 5-6, 7 years or more]	CS_13
14	The most prominent aspect of software quality that is associated with a CubeSat mission success is: [Reliability, Portability, Modifiability, Usability, Simplicity]	CS_14
15	The least prominent aspect of software quality that is associated with a CubeSat mission success is: [Reliability, Portability, Modifiability, Usability, Simplicity]	CS_15
16	Which of the following methodologies is most suitable for your CubeSat project? and why? [Waterfall, V-Model, SCRUM, Other]	CS_16
17	Which technique is best used for requirements elicitation? [Interview, Focus Group, Prototyping, Other]	CS_17
18	If you are given a second chance to redesign your CubeSat, what would you do instead? and why?	CS_18
19	CubeSat mission fail due to failure of tools rather than models.	CS_19
20	CubeSat mission fail due to failure of models rather than tools.	CS_20
21	Project scope and goals were clearly identified at the early stage of development.	CS_21
22	A methodology was carefully selected for a CubeSat project.	CS_22
23	Stakeholders can add/delete requirements in the CubeSat testing phase.	CS_23
24	Stakeholders participated heavily in the CubeSat design phase.	CS_24
25	A project team size has a positive effect on the CubeSat mission.	CS_25
26	Requirements were complex and hard to follow.	CS_26
27	Requirements were elicited and well documented.	CS_27
28	Requirements specification contains several non-testable functional requirements.	CS_28
29	The application of methodology has a positive effect on a CubeSat mission.	CS_29
30	Applied methodology enhanced the communication & control subsystems.	CS_30
31	A CubeSat project had attended to its original scope.	CS_31
32	A CubeSat project schedule was well managed.	CS_32

Since the mission result data is binary (the success or failure of a CubeSat), logistic regression was used to investigate the relationship between the categorical response variables. Logistic regression is, thus, the main statistical method used for this study. However, for complex survey responses with stratification and unequal weighting, "Proc Survey Reg" was applied to produce the appropriate estimates and standard error [100]. Further, "Proc Survey Reg" was used as a statistical tool for analyzing the survey data to test the relationship between 32 independent variables and the dependent variable (success or failure) of the CubeSat mission's success or failure.

5.1. Textual Data Analysis

The collected textual data (open-ended questions) from 127 respondents were analyzed and coded using Windows NVivo12. The data was stored as nodes (cases) and responses to each question were grouped together under a single node. Thematic content analysis, shown as a word cloud in Figure 5.1, was used to identify the words most frequently entered by the participants. For example, "mission goals" was entered more frequently than "tools" and "time". This is an indication of possible themes and the perspective of respondents. Figures 5.2 to 5.6 show thematic analysis and coding for the collected open-ended responses.

Figure 5.1. Word cloud for defining successful mission.

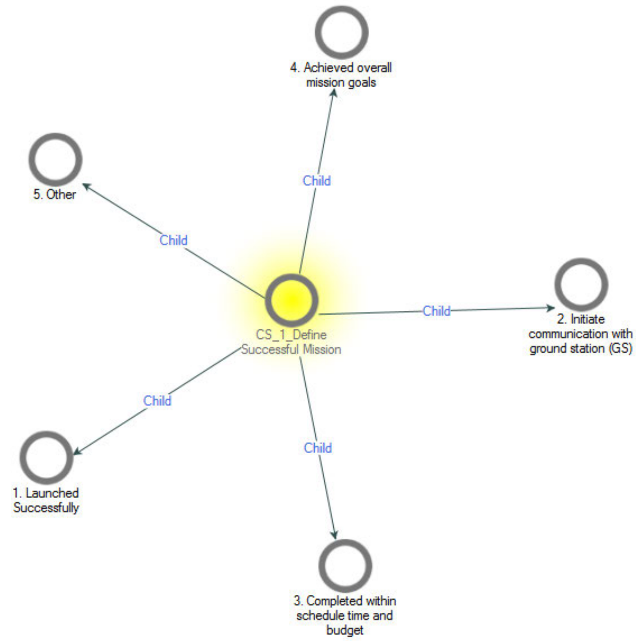


Figure 5.2. Coded nodes for defining successful mission (CS_1).

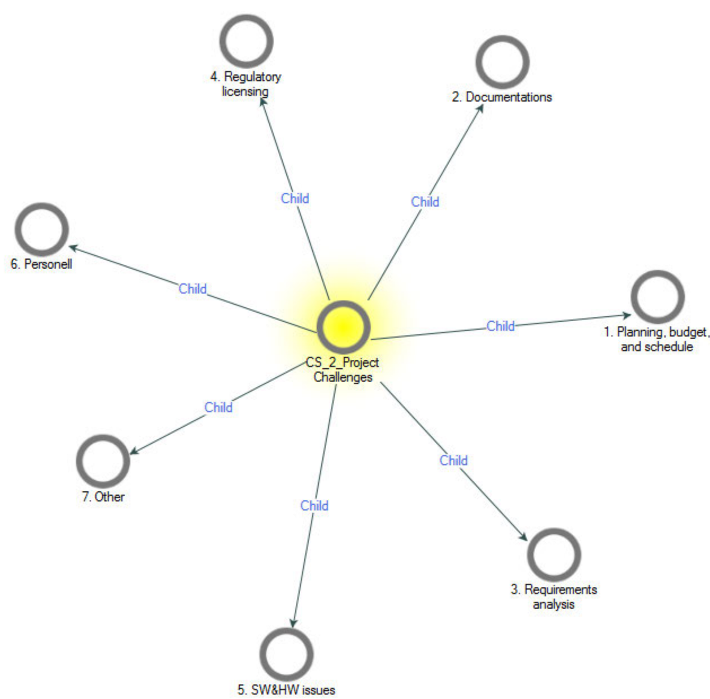


Figure 5.3. Coded nodes for project challenges (CS_2).

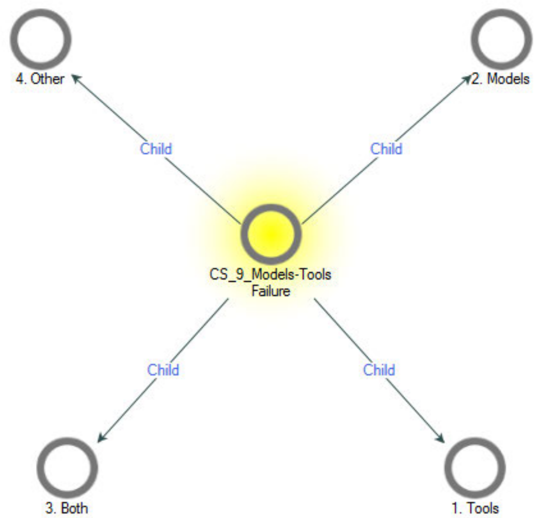


Figure 5.4. Coded nodes for models-tools failure (CS_9).

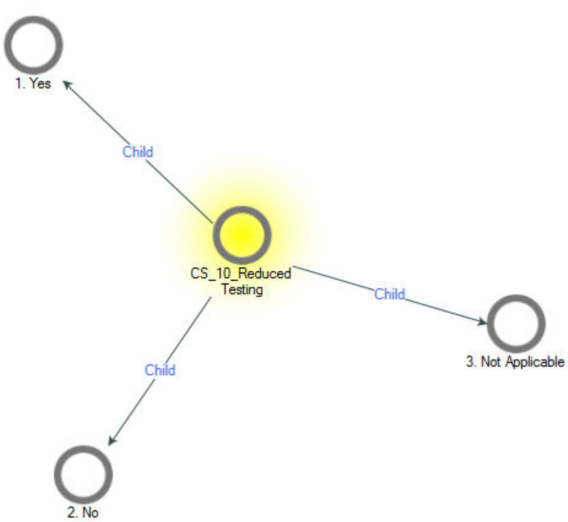


Figure 5.5. Coded nodes for reduced testing (CS_10).

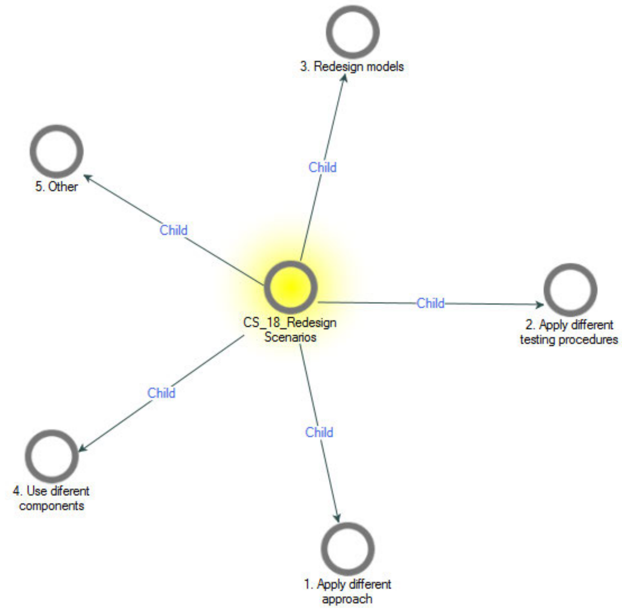


Figure 5.6. Coded nodes for redesign scenarios (CS_18).

The coded data was aggregated with the rest of the data for statistical analysis using SAS. Textual data from CS_1, CS_2, CS_9, CS_10, and CS_18 was coded and is presented in Tables 5.2 and 5.6. Respondents were asked questions that help in identifying common factors to determine the different perspectives of CubeSat developers towards CubeSat mission success. The first question was:

CS_1: How do you define a successful CubeSat mission?

Table 5.2. Define successful mission (CS_1).

	Frequency
Launched Successfully	25
Initiate communication with ground station (GS)	26
Completed within schedule time and budget	9
Achieved overall mission goals	51
Other	16

Answers for this question varied depending on the expertise and profession of the respondent. Therefore, thematic analysis was applied to help identify the common factors among the respondents' replies. Figure 5.7 shows the frequency of respondents identifying various definitions of a successful CubeSat mission.

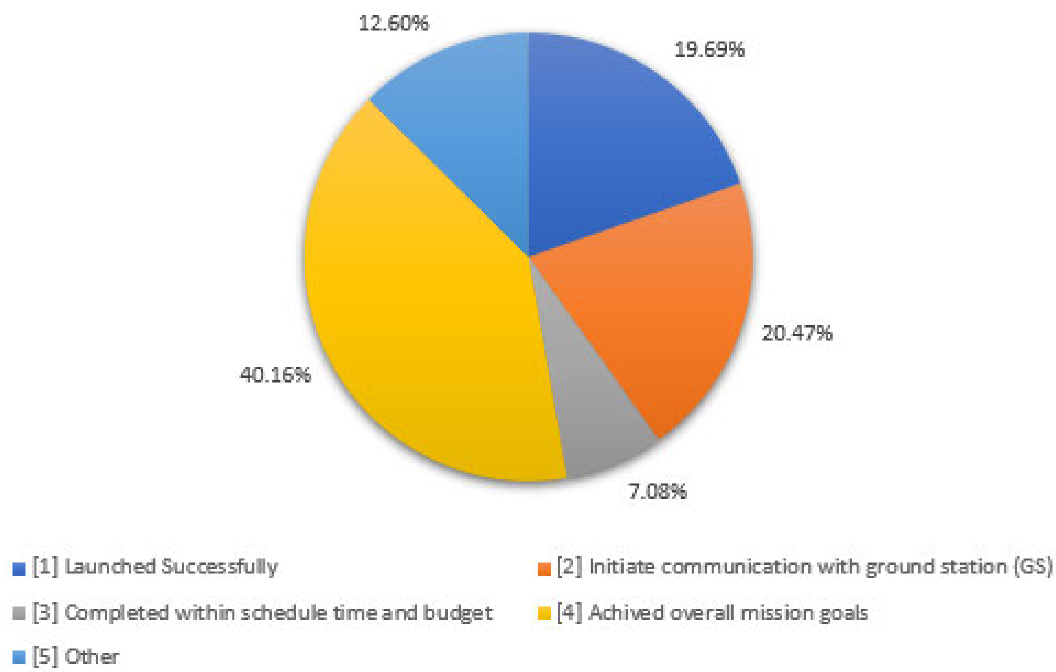


Figure 5.7. Frequency of responses for defining successful mission (CS_1).

The results from this analysis demonstrated the different perspectives of CubeSat developers towards CubeSat mission success. Defining a successful mission is a critical and important factor in ensuring mission success. Similarly, outlining mission objectives is key to quantifying the value of objectives to meeting strategic CubeSat mission standards. It also helps to understand how mission objectives are met and what factors impact mission objective attainment.

Answers varied based on the expertise and profession of the surveyed individual. Individuals with low to no experience in developing CubeSats defined success as having the project completed within schedule and budget. On the other hand, individuals with more experience in developing CubeSats defined success as achieving overall mission objectives, from planning to launching the CubeSat and receiving data. Here is an example of an open-ended response to CS_1: "each one is unique; you must define your objectives from day one. Some mission are successful if they make it on time to the launch pad, others need to generate a significant amount of revenue. It's the goal that you define on day one and revise/review along the way that defines your success."

Some of the respondents defined mission success as "other." this may be due to the significantly different perceptions and responses from respondents who identified as being from particular engineering disciplines, versus students and government personnel who regulate and license CubeSat projects. The role of a CubeSat developer, level of experience, and the mission status is also discussed in the descriptive analysis in section 5.3.

Respondents were also asked a second question to help in identifying common factors and to determine the different perspective of CubeSat developers toward CubeSat mission success. This question is stated as:

CS_2: What are the most challenging activities in a CubeSat development lifecycle?

Table 5.3. Challenging activities in a CubeSat project (CS_2).

	Frequency
Planning, budget, and schedule	20
Documentation	11
Requirements analysis	6
Regulatory licensing	4
SW/HW issues = design, testing, and integration	35
Personnel	24
Other	27

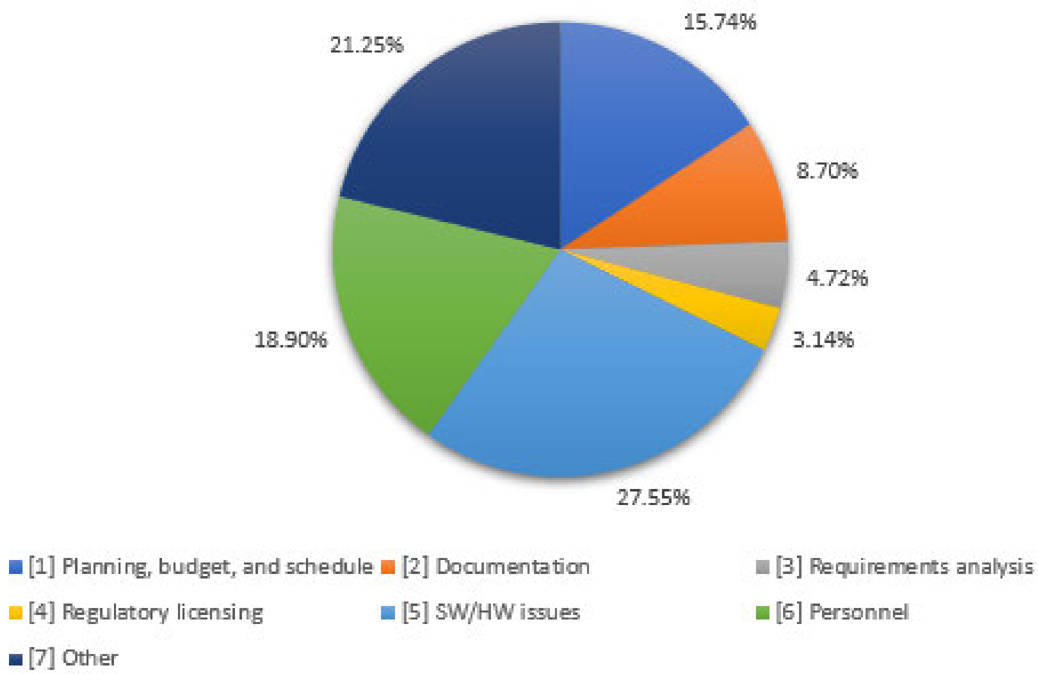


Figure 5.8. Frequency of responses for project challenges (CS_2).

The most frequent response to this question, as presented in Table 5.3 and Figure 5.8, is that project challenges mostly occur due to software and hardware issues. These issues were 27% of total responses. This factor is related to the design, testing, and integration of CubeSat components. One illustrative response, related to software and hardware issues, is: "The interfaces between the different units are not great. For example, there is i2c, can, LVDS, USB, Spacewire, etc. Each of them has a different set of problems and missions always have to do trade-offs on which one to choose for the different units. There should be "one" interface which allows you to connect any device. I hope that the Industry will eventually realize that USB 3.0 should be that interface."

Respondents also mentioned that there is typically a serious issue with power, as there is not enough area on the surface of the CubeSat for sufficient power generation. Further, deployable arrays add significant complexity to satellite design. Respondents also pointed out that it is always better to decide on tools early, rather than leaving these decisions until the last phase of development. They also suggested to incorporate only the necessary functionality during the design phase.

Respondents also noted challenges with the software used in their projects. Some of the problems included the flight software design being ignored during the CubeSat mission design, no debugging support, and no proper integration and testing processes. For personnel issues, respondents cited miscommunications between members, having no training for students, and some team members leaving the project before the implementation phase. Finally, the responses coded under "other" included some answers that are not responsive to CS_2.

Respondents were also asked if failures occur due to tools rather than models, or more failures occur in models than in tools. Expected answers included a discussion of failures that respondents have experienced in both tools and models. This data will help to identify reasons for failure in specific areas of CubeSat development. The question was stated as:
CS_9: Do most of system designs fail due to tools or models? Why?

Table 5.4. Reasons for CubeSat failure (CS_9).

	Frequency
Null	8
Tools	31
Models	47
Both	22
Other	19

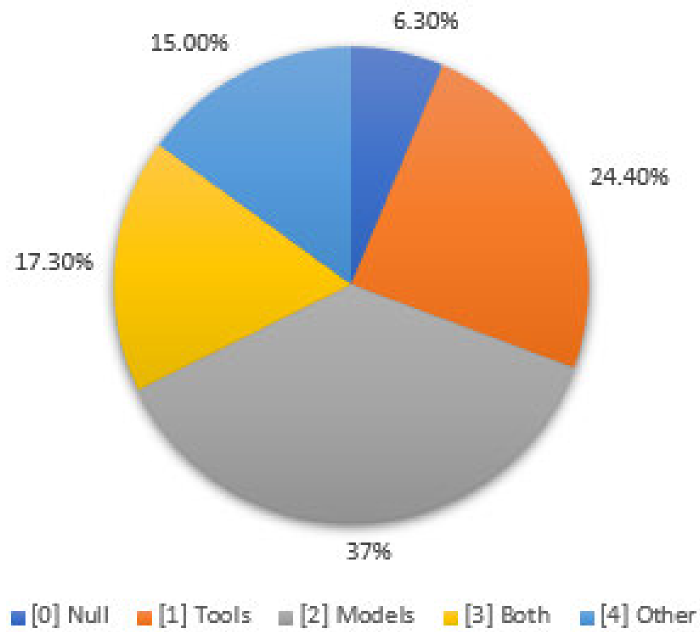


Figure 5.9. Frequency of responses for models-tools failure (CS_9).

As shown in Table 5.4 and Figure 5.9, survey respondents indicated that CubeSat project failures occur due to both the models and tools used in the project. However, most respondents indicated that CubeSat system failures occur due to the failure of models. Some mentioned that there was no proper methodology to follow. Others mentioned that, due to the restrictions of size and weight of CubeSats, adding new features and changing requirements led to the failure of models. Respondents also indicated that there are no standardized models used in the design phase of the project. On the other hand, respondents who think that CubeSat systems' failures are due to tools refer to unhardened COTS component failures or the functionality of such components affecting the communications between subsystems. For instance, using a thermo-electric cooler requires more power than a typical CubeSat can supply. Some responses, coded under "other", indicated that CubeSat failure is not always due to tools or models, but also attributed to other factors such as miscalculations or incorrect hardware assembly. Reasons for failure due to models, tools, or both are listed below:

"No flight software is seriously included in CubeSat design"

"Thermal analysis is the least to be included in CubeSat design"

"CPUs are not latch-up protected"

"The microcontrollers have limited computational power"

"Issues encountered with STM32"

"TMTC failure"

"No FDIR approach"

"No FMECA analysis"

"Limited functionalities of COTS"

"No debugging supports"

"Complexity of flight software systems architecture"

"CFS suite from NASA is needed"

"VHF/UHF transceivers failure"

"Communication problems with simplex and duplex radio"

"Structure and deployable design (STR) "

Respondents were asked to identify best practices that can help in design improvement or in reducing testing requirements. This includes identifying steps and processes to avoid in the design phase. The following question asked respondents if they believe that their CubeSat design had reduced the testing efforts required in the testing phase or if their design had improved performance overall. It was stated as: CS_10: Has your system design reduced testing efforts and/or improved performance?

Table 5.5. Reducing testing efforts and improve performance (CS_10).

	Frequency
Null	14
Yes	16
No	76
N/A	21

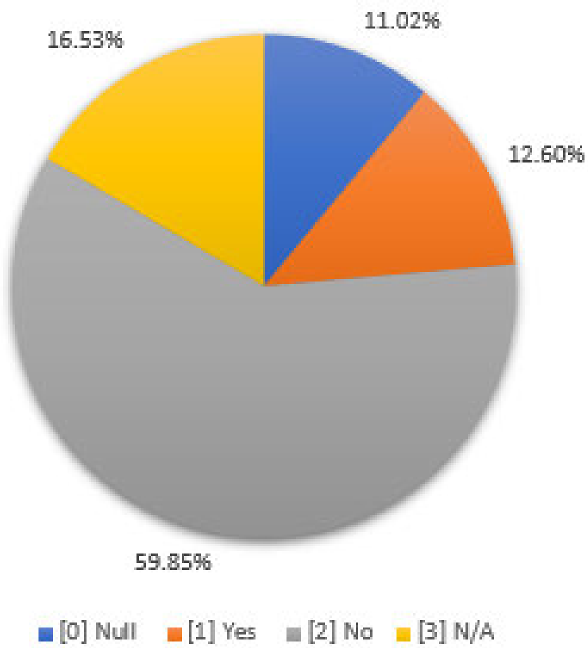


Figure 5.10. Frequency of responses for reduced testing (CS_10).

Figure 5.10 and Table 5.5 show that approximately 17% of respondents either do not know if the CubeSat they have designed had reduced testing time requirements or not. This may be because they have not reached the testing phase yet, they lack something to compare to, or the respondent is not involved in the testing process. However, 60% of respondents think that their design has not reduced the required testing time. This suggests that traditional methodologies require more testing time for space systems and that more testing should be performed on such systems. This, however, does not necessarily mean that their CubeSat projects have failed. Additionally, those who claimed that their design helped reduce testing time suggested that following verification and validation (V&V) methodologies and performing TVAC testing can reduce not only the testing time but also improve system performance.

The next question asked respondents about how they would change practices in retrospect. The question was presented as:

CS_18: If you are given a second chance to redesign your CubeSat, what would you do?

Table 5.6. Redesign CubeSat System (CS_18).

	Frequency
Null	16
Apply different approach	20
Apply different testing procedures	10
Redesign models	21
Use different components	20
Other	40

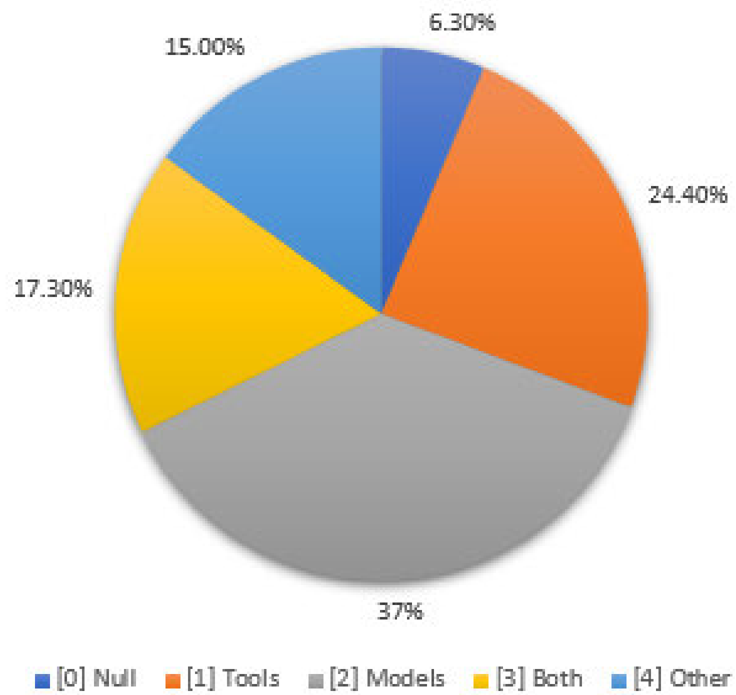


Figure 5.11. Frequency of responses for redesign scenarios (CS_18).

Figure 5.11 and Table 5.6 show that some respondents indicated that, if they had the chance to redesign their CubeSats, they would use different tools than the ones they have used. Alternatively, 31% of respondents, coded under "other", said that even if they were given the chance to redesign their CubeSat system, they would not apply any changes to their CubeSat project practices. Some of those who would change the tool that they used attributed faults to the unexpected performance of the COTS hardware. Others discussed the need to make their design more easily modifiable to be compliant with regulations. Some responses that discussed applying a different approach indicated that respondents were not aware of early project steps including planning, scheduling, budgeting, and selecting a specific methodology for the project.

Some have also suggested that following rapid development or Agile methodologies for their project would be good, since they have limited time and budget for their project, which can lead to project failure. Others, who preferred following different testing procedures, thought that some needed testing for their project was ignored, such as thermal testing. Respondents also mentioned that they used different models because they faced challenges with adding or removing features or components after the design phase of the project.

5.2. Testing Hypotheses

This section tests six hypotheses that were formed from the factors presented in the previous chapters. These factors affect CubeSats' mission success rate. Each factor includes sub-factors, based on survey questions. These questions relate to systems engineering processes and practices used in the CubeSat development lifecycle. Some of the questions, are included in multiple factors that share similarities in their characteristics. Table 5.7 lists the survey questions that are associated with each factor.

Table 5.7. Factors and hypotheses.

Factor	Code	Hypothesis	Survey Questions
Defining Mission Objectives	DMO	H_1	CS_1, CS_3, CS_16, CS_17, CS_21, CS_31
Timeline Analysis	TA	H_2	CS_3, CS_12, CS_13, CS_16
Mission Assurance	MA	H_3	CS_2, CS_10, CS_18, CS_16, CS_22, CS_23, CS_24
Critical Design Review	CDR	H_4	CS_2, CS_9, CS_10, CS_18, CS_4, CS_14, CS_15
Experience in Quality Attributes	EQA	H_5	CS_6, CS_14, CS_15, CS_16
Schematic Diagram	SD	H_6	CS_9, CS_11, CS_13, CS_16, CS_23, CS_24

The level of significance used in the study was 5%. All hypotheses used for this research were tested using "PROC SURVEYREG" in SAS version 9.4. Since the p-values for all factors were less than the level of significance (P<0.05), all factors influencing CubeSat mission success were deemed to be significant. Each factor was based on multiple variables in the study.

5.2.1. Defining Mission Objectives

CubeSat development teams may never discuss the scope and objectives of their mission; however, that does not always lead to failure of the mission. Nevertheless, the study measured the number of CubeSat missions that have succeeded which clearly defined their mission at an early stage in the project's life cycle. The hypothesis statement derived from this factor is:

H_1: Clearly defining mission objectives (DMO) at an early stage of a CubeSat project's life cycle has a positive effect on the mission's success rate.

Defining mission objectives guides the developers during the entire project and acts as a point of reference impacting the success of the project. DMO was assessed based on six questions relevant to mission definition. These include the perception of development status and the requirements of the system. The results indicate that the developer's ability to identify the project's scope and goals, at the early stages of development, had a significant influence on the success of the project. Table 5.8 displays the test for model effects of DMO.

Table 5.8. Tests of model effects of DMO.

Effect	Num DF	F Value	Pr > F
Model	6	6.28	< .0001
Intercept	1	3.11	0.0814
Define successful mission	1	0.54	0.4654
CubeSat mission type	1	0.16	0.6891
Methodology Type	1	4.82	0.0309
Requirements elicitation techniques	1	1.30	0.2576
Project scope and goals	1	23.45	< .0001
CubeSat project had attended to its original scope.	1	0.33	0.5680

From Table 5.8, it can be observed that the p-value of the F-test is 0.0001, which is less than the 5% significance level. Based on the survey results, defining mission objectives (DMO) has positive effects on the mission's success rate.

5.2.2. Timeline Analysis

Testing time needs may vary due to the complexity of a mission. A goal of this study is to determine if a CubeSat's design helped in reducing the required testing time. Thus, using a timeline analysis technique, which is a critical systems engineering process, in a CubeSat project will clarify the relationships between functions and tasks, and will identify the specific time allocated for design requirements. Timeline analysis can show the required time needed, as well as the design constraints in place, while building CubeSat systems. The hypothesis derived from this factor is:

H_2: Implementing a timeline analysis (TA) in a CubeSat project has a positive effect on the mission's success rate.

There are four questions used to assess this factor. The questions relate to the development phase of the system, the type of the developed CubeSat, aspects of software quality, and factors associated with failure and success of the CubeSat project. Table 5.9 displays the test for model effects of TA.

Table 5.9. Tests of model effects of TA.

Effect	Num DF	F Value	Pr > F
Model	4	67.57	< .0001
Intercept	1	395.50	< .0001
CubeSat mission type	1	0.50	0.4825
CubeSat type	1	161.74	< .0001
Time needed for development	1	0.02	0.9015
Methodology type	1	34.68	< .0001

From Table 5.9, it can be observed that the p-value of the F-test is 0.0001, which is less than the 5% significance level. Based on the survey results, implementing timeline analysis (TA) in a CubeSat project has a positive effect on the success rate of a CubeSat mission.

5.2.3. Mission-Assurance Analysis

Mission assurance depends on mission risk management, which requires mission goals to be defined. Running a risk assessment test is very important to ensuring a mission's success. This test will help in identifying flaws and issues at the early stages of CubeSat development. However, it is also important to know if developers have used or discussed risk-based practices. Tests may include, but are not limited to, communication link testing, power system testing, thermal vacuum testing (TVAC), and regular reviews (formal and informal). The hypothesis statement derived from this factor is:

H_3: Performing mission-assurance analysis (MA) in a CubeSat project life cycle has a positive effect on the mission's success rate.

The results indicate that performing mission-assurance analysis (MA), in a CubeSat project, has a positive effect on the mission's success rate. There were seven questions assessed for the MA factor. The questions related to designing and redesigning the system and the system's requirements. Table 5.10 displays the test for the model effects of MA.

Table 5.10. Tests of model effects of MA.

Effect	Num DF	F Value	Pr > F
Model	7	75.58	< .0001
Intercept	1	272.74	< .0001
Project Challenges	1	18.36	< .0001
Reduced testing	1	224.75	< .0001
Redesign scenarios	1	67.53	< .0001
Methodology type	1	40.57	< .0001
Methodology selection	1	3.00	0.0856
Stakeholders participating in the testing phase	1	33.42	< .0001
Stakeholders participating in the design phase	1	23.24	< .0001

From Table 5.10, it can be observed that the p-value of the F-test is 0.004, which is less than the 5% of significance level. Based on the survey results, performing mission-assurance analysis (MA) in a CubeSat project has a positive effect on the mission's success rate.

5.2.4. Critical Design Review

Complexity of system design is very common in CubeSat development. A simple design could be critical to a mission's success in some environments. Survey respondents were asked about best practices, their experience with design problems and issues, and if they could identify an alternate approach to avoid future mission failure.

A critical design review (CDR) is performed during the system development phase. It may include a series of reviews conducted for hardware and software systems, coding, and testing. Additionally, test plans are reviewed to assess if test efforts are developing sufficiently. The approved detailed design serves as the basis for final production planning and initiates the development of the final software code. The CDR may also include metrics for measuring design optimizations and constraints. The hypothesis statement derived from this factor is:

H_4: Performing a critical design review (CDR) in the design phase has a positive effect on the mission's success rate.

This factor was assessed using seven questions relevant to system design efforts, techniques for requirements elicitation, and the time needed for developing a typical CubeSat. Table 5.11 displays the test for model effects of CDR.

Table 5.11. Tests of model effects of CDR.

Effect	Num DF	F Value	Pr> F
Model	7	64.10	< .0001
Intercept	1	234.48	< .0001
Project Challenges	1	26.41	< .0001
Tools-Models failure	1	0.02	0.8782
Reduced testing time	1	217.60	< .0001
Redesign scenarios	1	49.38	< .0001
Team size	1	72.85	< .0001
Most aspect of software quality	1	6.05	0.0153
Least aspect of software quality	1	18.83	< .0001

From Table 5.11, it can be observed that the p-value of the F-test is 0.0001, which is less than the 5% significance level. Based on the survey results, performing a critical design review (CDR) during the design phase has a positive effect on the mission's success rate.

5.2.5. Experience in Quality Attribute Practices

The experience in quality attribute practices factor considers developers' experience with quality attribute practices including usability, security, and reliability. It also measures the correlation between mission success, having expert personnel on the team, the number of years a developer has spent in this environment, the number of CubeSat projects a developer has been involved with, and mission success. The hypothesis statement derived from this factor is:

H_5: Experience in quality attribute (EQA) practices has a positive effect on the mission's success rate.

Results indicated that experience in quality attribute practices has a positive effect on a mission's success rate. To assess the variables that influence EQA, four questions were used: The aspects of software quality most and least associated with mission success, the type of a CubeSat mission, and the time needed to develop a typical CubeSat. Table 5.12 displays the test for model effects of EQA.

Table 5.12. Tests of model effects of EQA

Effect	Num DF	F Value	Pr> F
Model	4	73.80	< .0001
Intercept	1	60.26	< .0001
Number of CubeSat projects	1	190.13	< .0001
Most aspect of software quality	1	13.36	0.0004
Least aspect of software quality	1	11.82	0.0008
Methodology type	1	35.46	< .0001

From Table 5.12, it can be observed that the p-value of the F-test is 0.0001, which is less than the 5% of significance level. Based on the survey results, experience in quality attribute (EQA) practices has a positive effect on the mission's success rate.

5.2.6. Schematic Diagram

The schematic diagram (SD) is a systems engineering process that can depict hardware and software components and their interrelationships. SDs are developed at successively lower levels as the analysis proceeds to define lower-level functions within higher-level requirements. These requirements are further divided and allocated using a requirements matrix (RM). The hypothesis derived from this factor is:

H_6: Including a schematic diagram (SD) in a CubeSat project has a positive effect on the mission's success rate.

Results indicate that including a schematic diagram (SD) in a CubeSat project has a positive effect on the mission's success rate. Hence, it is clear that the ability of a developer to determine the factor most associated with CubeSat project failure is essential for mission success. Table 5.13 displays the test for model effects of SD.

Table 5.13. Tests of model effects of SD.

Effect	Num DF	F Value	Pr> F
Model	6	23.06	$< .0001$
Intercept	1	240.46	$< .0001$
Tools-Models failure	1	8.56	0.0041
Factors of failure	1	0.02	0.8996
Time needed for development	1	0.36	0.5473
Methodology type	1	31.80	$< .0001$
Stakeholders participating in the testing phase	1	68.66	$< .0001$
Stakeholders participating in the design phase	1	29.25	$< .0001$

From Table 5.13, it can be observed that the p-value of the F-test is 0.0001, which is less than the 5% significance level. Based on the survey results, including a schematic diagram (SD) in a CubeSat project has a positive effect on the mission's success rate. Including a SD in a CubeSat project is expected to help ensure that a robust CubeSat system results. However, flaws in a SD may lead to overall mission failure. Knowing where defects in models exist and how to fix them is also important. Including a SD in a CubeSat project provides visibility of related system components and traceability to the requirements matrix and other system engineering documentation.

5.3. Descriptive Data Analysis

This section presents cross-tabulation analysis that shows the association between frequency tables. Correlation between the status of the CubeSat project and the type of the CubeSat is also discussed. Additionally, this section discusses the association between the type of mission and the applied methodologies, and the association between the type of CubeSat and the time needed for development. Data from survey respondents is included in Table 5.14. The data shows that respondents participated or are participating in 28 successful, 11 failed, 48 incomplete, and 35 under-development CubeSat projects.

Table 5.14. Status of CubeSat project.

	Frequency
Failed	11
Incomplete	48
Succeed	28
Under-development	35
Frequency Missing = 5	

Table 5.15 and Figure 5.12 show the number of successful CubeSat projects for various CubeSat types. Table 5.15 also shows successful CubeSat projects associated with their types.

Table 5.15. Association between a CubeSat Success and its type.

	CubeSat type								
Type	1U	1U 2U	1U 2U 3U	1U 3U	2U	3U	3U Other	Other	Total
Successful	10	2	1	7	1	5	1	1	28

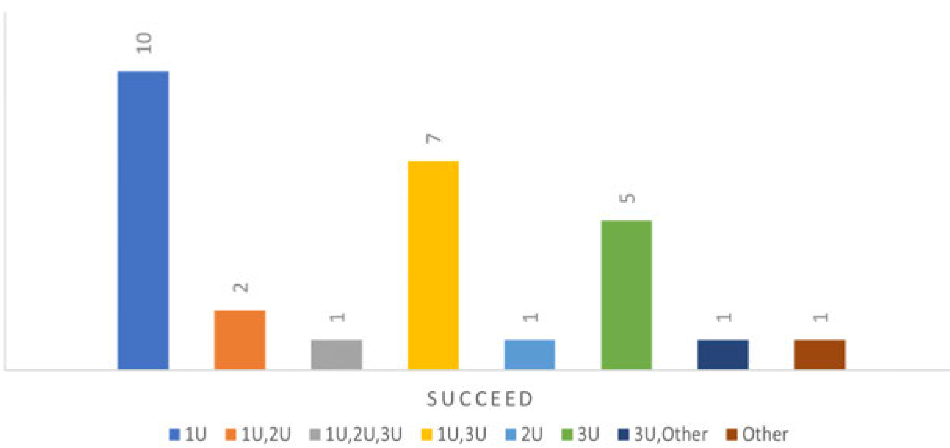


Figure 5.12. Successful CubeSats by type.

The data presented in Table 5.15 and Figure 5.12 includes only successful CubeSats and their types. The majority of successful CubeSats are 1U type. Note that, in some cases, respondents indicated that they have developed two or more types of CubeSats. Additionally, the other category "other" includes types such as 6U or 27U.

Table 5.16. Association between a CubeSat failure and its type.

	CubeSat type		
Type	1 U	3 U	Total
Failed	6	5	11

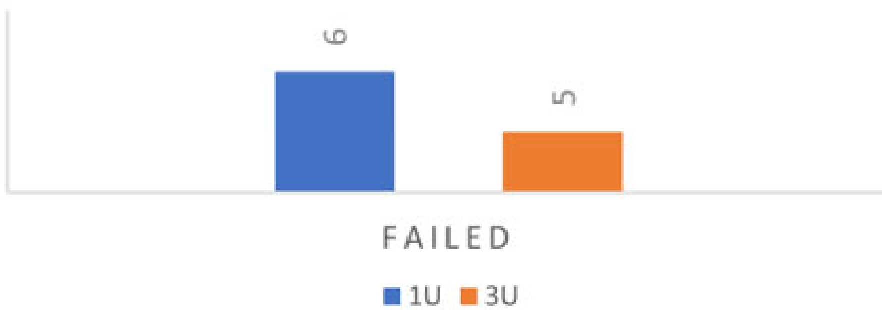


Figure 5.13. Failed CubeSats' type.

Data presented in Table 5.16 and Figure 5.13 indicates that most CubeSat failures occur in 1U and 3U types (9% of responses). This, however, does not mean that failure is frequent in these CubeSat types, since they also are the most frequently developed CubeSat types, accounting for 78% of total successful CubeSats. A high failure rate may occur among student projects, which may have a correlation with CubeSat size.

Table 5.17. Association between CubeSat failure and the reason for failure.

Status of CubeSat project	Models/Tools Failure					
Frequency	0	1	2	3	4	Total
Failed	1	3	3	3	1	11
Total	1	3	3	3	1	11

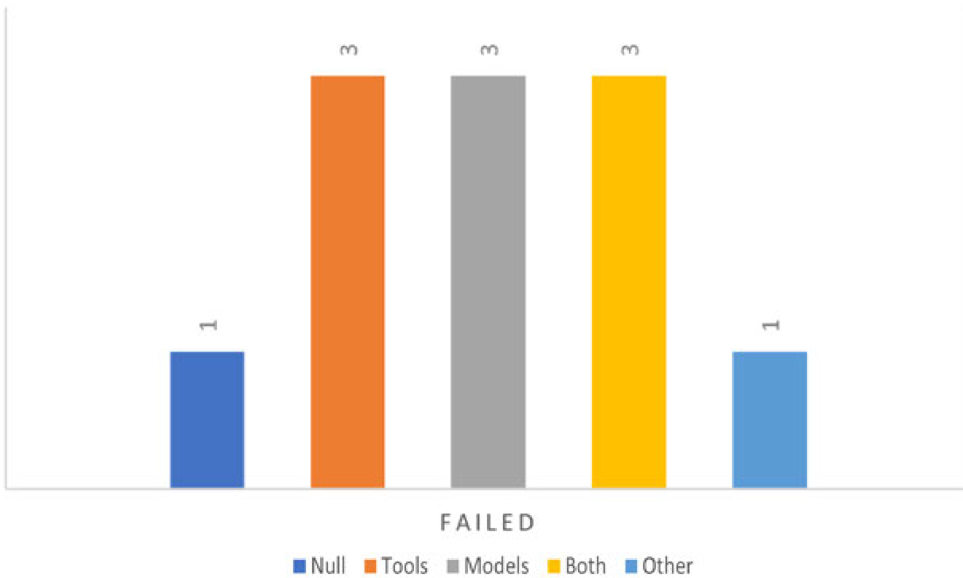


Figure 5.14. CubeSat failure and the reason for failure.

Respondents indicated that CubeSat project failures occur due to the models and tools used in the project. Some also mentioned that there was no proper methodology to follow. Others mentioned that the restrictions regarding the size and weight of the CubeSat, adding new features, and changing requirements lead to the failure of models. Respondents also indicated that there were no standardized models used in the design phase of the project. Measuring the association between the failure of CubeSats and the reasons for failure, as presented in Table 5.17, is important. However, due to limited number of respondents who indicated the failure of their CubeSats (less than 10% of responses), it is difficult to determine whether failure occurs more due to issues with models than due to tool issues or vice-versa. Data presented in Table 5.17 and Figure 5.15, shows that most respondents' CubeSat missions fall under the educational category (66% of total responses).

Table 5.18. Association between CubeSat missions and their types.

Type of CubeSat mission	CubeSat type								
	1U	1U,2U	1U,2U,3U	1U,3U	2U	3U	3U,Other	Other	Total
Commercial	0	0	0	1	1	1	0	0	3
Educational	46	7	0	2	3	23	0	3	84
Other	0	0	1	0	0	1	0	0	2
Scientific	6	0	0	3	1	6	0	1	17
Scientific and Educational	7	0	0	5	1	0	0	1	14
Scientific and Educational and Commercial	1	0	0	0	0	0	1	0	2
Total	60	7	1	11	6	31	1	5	122
Frequency Missing = 5									

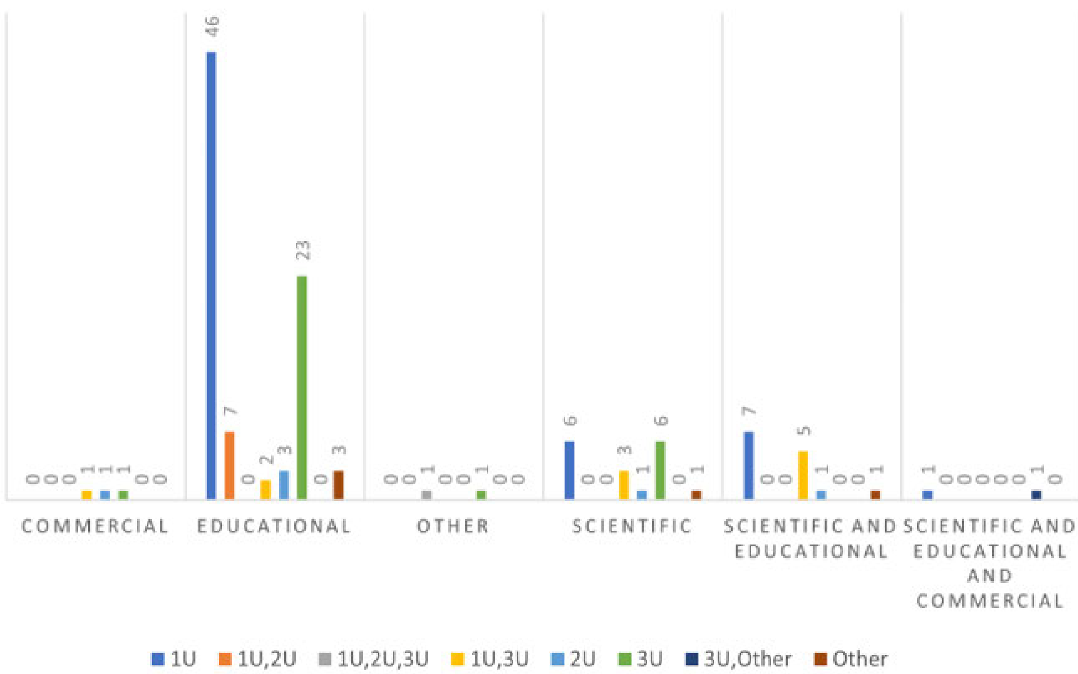


Figure 5.15. CubeSat missions and their types.

Data presented in Table 5.19 and Figure 5.16, shows that the methodology most used in CubeSat projects is the V-model (49% of total responses). This is expected since the V-model is widely used for space systems [100]. The SCRUM model was the second most frequently used methodology that respondents indicated (25% of total responses). SCRUM is a type of Agile methodology.

Table 5.19. CubeSat missions and their association with applied methodologies.

Type of CubeSat mission	Methodology				
	Other	SCRUM	V-Model	Waterfall	Total
Commercial	1	0	2	0	3
Educational	6	22	43	13	84
Other	0	0	2	0	2
Scientific	3	3	6	5	17
Scientific and Educational	1	6	7	0	14
Scientific and Educational and Commercial	1	0	0	1	2
Total	12	31	60	19	122
Frequency Missing = 5					

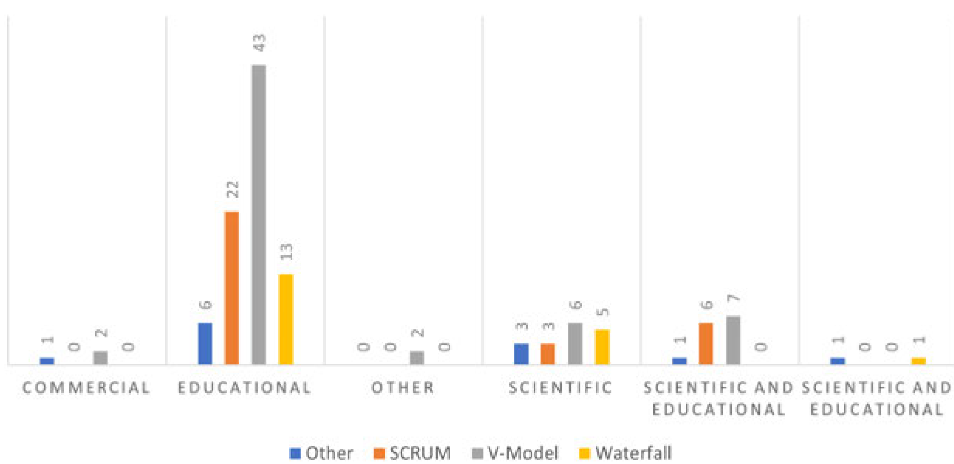


Figure 5.16. CubeSat missions and their association with applied methodologies.

Data presented in Table 5.20 and Figure 5.17 shows that the most typical timeframe for 1U and 3U CubeSats is 2 years or less. Roughly 20% of respondents indicated that it may take 3 to 4 years to develop a CubeSat. A minimal number of respondents indicated that developing a CubeSat may take more than 5 years.

Table 5.20. CubeSat types associated with typical time needed for development.

CubeSat type	Typical time needed			
	2 years or less	3-4 years	5-6 years	Total
1U	52	8	0	60
1U,2U	6	1	0	7
1U,2U,3U	1	0	0	1
1U,3U	10	1	0	11
2U	4	2	0	6
3U	21	9	1	31
3U,Other	1	0	0	1
Other	1	4	0	5
Total	96	25	1	122
Frequency Missing = 5				

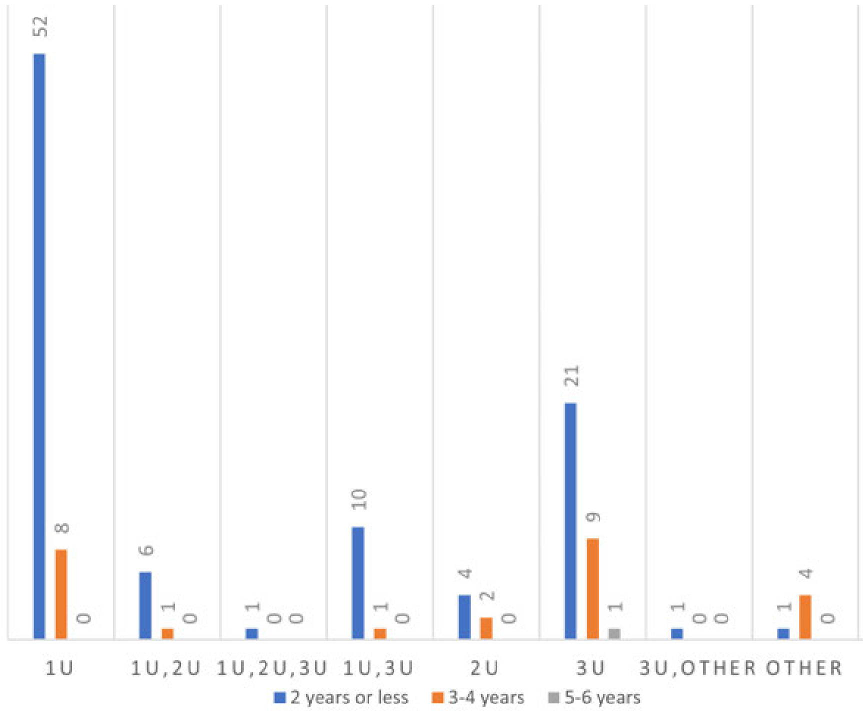


Figure 5.17. CubeSat types associated with typical time needed for development.

Data in Table 5.20 and Figues 5.17, shows that 78% of respondents indicated that developing a CubeSat requires two years or less of development time. This timeframe is comprised of multiple phases. Some run concurrent with others. The concept phase, for instance, may take up to six months, and must be completed before other phases, whereas developing and testing the ground station may take a whole year but can be done concurrent with other activities.

5.4. Likert Scale Analysis

The section presents analysis related to survey questions 19 to 32, which use a Likert Scale. Each question asks respondents to select the response that best characterizes their views regarding the presented statement. Response options were: 1 for strongly disagree, 2 for disagree, 3 for neither agree nor disagree, 4 for agree, and 5 for strongly agree.

The first question in this section asks respondents about the attribution of CubeSat failures to tools versus models. This question was presented as:

CS_19: CubeSat mission fail due to failure of tools rather than models.

Table 5.21. CubeSat mission and reasons for failure.

	Frequency
Agree	52
Disagree	41
Neither agree nor disagree	8
Strongly agree	17
Strongly disagree	4
Frequency Missing = 5	

Based on the data shown in Table 5.21, it is evident that many respondents (40.94%) believe that CubeSat missions fail due to unsatisfactory tools rather than models. Others disagreed with the statement (32.28%). Respondents who indicated that CubeSat systems fail due to tools refer to unhardened COTS or the functionality of such components affecting the communications between subsystems. Respondents noted that problems with tools may include the complexity of flight software, systems architecture, or that no debugging support exists.

The second question in this section asked the reverse of the first question. This was presented as:

CS_20: CubeSat mission fail due to failure of models rather than tools.

Table 5.22. CubeSat mission and reasons for failure.

	Frequency
Agree	41
Disagree	46
Neither agree nor disagree	11
Strongly agree	23
Strongly disagree	1
Frequency Missing = 5	

The data presented in Table 5.22 suggests that, while some respondents (36.22%) disagreed with the notion that CubeSat missions fail largely due to issues with models instead of tools, others (32.28%) agreed with this concept. Respondents noted that the selection of a methodology may lead to project failure. Other respondents mentioned that the restrictions on the size and weight of the CubeSat, adding new features and changing requirements lead to the failure of models. Respondents also added that there were no standardized models used during the design phase of projects.

The third question in this section asked about the project's scope and goals. This was presented as:

CS_21: Project scope and goals were clearly identified at the early stage of development.

Table 5.23. Identified scope and goals for the project.

	Frequency
Agree	65
Disagree	31
Neither agree nor disagree	15
Strongly agree	9
Strongly disagree	2
Frequency Missing = 5	

Based on the data presented in Table 5.23, most respondents (51.18%) indicated that the project scope and goals were clearly identified at the beginning of the project. Others (24.41%) disagreed with this statement. This indicates that respondents were aware of the need for identifying scope and goals prior to the design and testing phases of the project. Additionally, the developer's ability to identify the project's scope and goals at the early stages of development had a significant influence on the success of the project.

The fourth question in this section relates to project methodology. It was presented as:

CS_22: A methodology was carefully selected for a CubeSat project.

Table 5.24. Selection of methodology.

	Frequency
Agree	56
Disagree	37
Neither agree nor disagree	17
Strongly agree	8
Strongly disagree	4
Frequency Missing = 5	

Table 5.24 shows that many respondents (44%) indicated that a methodology was carefully selected for their projects. Respondents noted, though, that the selection of a methodology may lead to project failure. Lots of other factors were mentioned including that restrictions on the size and weight of the CubeSat, adding new features and changing requirements lead to the failure of models. Some respondents also indicated that there were no standardized models used in the design phase of the project.

The fifth question in this section considers the role of stakeholders in a CubeSat project. It is presented as:

CS_23: Stakeholders can add/delete requirements in the CubeSat testing phase.

Table 5.25. Stakeholders engagements in the testing phase.

	Frequency
Agree	52
Disagree	44
Neither agree nor disagree	12
Strongly agree	5
Strongly disagree	9
Frequency Missing = 5	

Table 5.25 demonstrates that many (41%) respondents indicated that they believe that stakeholders could add or delete requirements in the testing phase for a project. However, 35% of respondents disagreed with this statement. This suggests that most respondents are open to modification of requirements in the testing phase. Only 7% of respondents strongly disagreed with this statement. Respondents noted that if stakeholders can add or delete requirements, they will be satisfied with developed system; however, this may add more time and cost to the development process. Alternately, those indicating opposition to stakeholder changes during the development process indicated that they were following traditional methodologies such as V-model or waterfall, where it is difficult to go back to previous phases once a phase is complete. These approaches may increase risk and lead to project failure.

The sixth question in this section asked respondents about the level of stakeholder involvement in their project. It was presented as:

CS_24: Stakeholders participated heavily in the CubeSat design phase.

Table 5.26. Stakeholders engagements in the design phase.

	Frequency
Agree	50
Disagree	39
Neither agree nor disagree	21
Strongly agree	7
Strongly disagree	5
Frequency Missing = 5	

Table 5.26 presents the data from question CS_24, that asks respondents whether stakeholders participated heavily in the design phase. Some respondents (39.37%) said that they did, while 30.70% of respondents said they didn't. Respondents who agreed with this statement indicated that designing a preliminary prototype to show to stakeholders helps in detecting errors early in the design process and helps in reducing cost and time. It is also helps in creating stakeholder satisfaction. Alternately, those who didn't have stakeholders engage in the design phase indicated

that this only works with certain projects when requirements are divided up into small iterations and design can focus on one requirement at a time. Some respondents indicated that, after completing the design phase and other tasks, identifying changes in the testing phase requires a difficult transition back to the design phase to alter requirements to meet changed stakeholder needs.

The seventh question in this section considers the effect of team size on mission success. This question is presented as:

CS_25: A project team size has a positive effect on the CubeSat mission.

Table 5.27. Project team size effect.

	Frequency
Agree	68
Disagree	26
Neither agree nor disagree	12
Strongly agree	15
Strongly disagree	1
Frequency Missing = 5	

Table 5.27 explores whether the size of a project team has an effect on mission success. Most (53.54%) respondents said that it did, while 20.47% said that it did not. While most respondents indicated that a larger team would positively enhance a project, other respondents indicated that smaller teams are better. Respondents noted that CubeSats developed for educational purposes typically have limited team sizes, timeframe, and budgets. The team size can be a major factor in software-related CubeSat project failure, since smaller teams usually have limited software engineering capabilities.

The eighth question in this section discusses requirements complexity and difficulty of understanding. This was presented as:

CS_26: Requirements were complex and hard to follow.

Table 5.28. Complexity of requirements.

	Frequency
Agree	59
Disagree	40
Neither agree nor disagree	15
Strongly agree	8
Frequency Missing = 5	

Table 5.28 presents data regarding whether respondents thought that mission requirements were complete and difficult to follow. Many respondents (46.45%) indicated that they were. While others (31.49%) didn't experience this difficulty. Many respondents indicated that they see requirements as a big issue in CubeSat development stages, as some requirements are hard to follow due to ambiguity and perhaps a lack of experience.

The ninth question in this section dealt with requirements' source and documentation. This was presented as:

CS_27: Requirements were elicited and well documented.

Table 5.29. Requirements elicitation and documentation.

	Frequency
Agree	70
Disagree	22
Neither agree nor disagree	16
Strongly agree	10
Strongly disagree	4
Frequency Missing = 5	

Table 5.29 presents data regarding whether project requirements were elicited and well-documented. Most (58.33%) respondents agreed that they were, while 17.32% disagreed. This

does not contradict CS_26, as finding requirements complex or hard to follow does not necessarily suggest that requirements were not well elicited or documented. A lack of proper documentation, though, may potentially lead to problems.

The tenth question in this section dealt with requirement testability. This was presented as:

CS_28: Requirements specification contains several non-testable functional requirements.

Table 5.30. Functional requirements specification.

	Frequency
Agree	50
Disagree	39
Neither agree nor disagree	9
Strongly agree	11
Strongly disagree	12
Frequency Missing = 6	

Table 5.30 presents data regarding whether project specifications contained non-testable functional requirements. May respondents (41.66%) indicated that they did, while 32.50% of respondents indicated that they did not.

The presence of non-testable requirements is particularly relevant to the potential utility of a CubeSat system, as the testing phase is an essential part of the process. The fact that most respondents believe there to be non-testable functional requirements means that their systems cannot be fully tested. These requirements must be modified to be testable in order for testing to be effective. The project team must be confident in their project requirements and their testability. Deviation from this may result in wasted time, effort and resources being spent trying to test unsuitable functions.

The eleventh question in this section considers the impact of methodology on CubeSat mission success. This was presented as:

CS_29: The application of methodology has a positive effect on a CubeSat mission.

Table 5.31. Methodology effects on a CubeSat mission.

	Frequency
Agree	71
Disagree	24
Neither agree nor disagree	10
Strongly agree	14
Strongly disagree	3
Frequency Missing = 5	

Table 5.31 presents data regarding whether the application of a methodology to the project has a positive impact on CubeSat missions. Over half (55.90%) of respondents indicated that it did, while 18.89% expressed disagreement. These results reiterate that each of the previously discussed components of a project is essential to completing the project. Failure to use an appropriate methodology may cause deviation from the core objectives of the project and even project failure.

The twelfth question in this section considers the impact of methodology on communications and control. This was presented as:

CS_30: Applied methodology enhanced the communication and control subsystems.

Table 5.32. Methodology effects on CubeSat subsystems.

	Frequency
Agree	74
Disagree	19
Neither agree nor disagree	13
Strongly agree	13
Strongly disagree	3
Frequency Missing = 5	

Table 5.32 presents data regarding whether a methodology enhanced a CubeSat's communication and control subsystems. Of the total responses, 58.26% indicated that it did, while 14.96% disagreed. These results reiterate the findings of CS_29 regarding the importance of requirements. While respondents appear to believe that methodology selection is beneficial for the overall project, they also appear to believe it to be particularly useful for the communication and control subsystems.

The thirteenth question in this section relates to project scope. This was presented as:

CS_31: A CubeSat project had attended to its original scope.

Table 5.33. Original scope of the project.

	Frequency
Agree	77
Disagree	27
Neither agree nor disagree	11
Strongly agree	5
Strongly disagree	2
Frequency Missing = 5	

Table 5.33 presents data indicating whether each CubeSat project had achieved its original scope. A majority of respondents (60.62%) expressed that projects did meet their scope requirements, while 21.26% said that they did not. Notably, respondents may have also disagreed with the statement because their project lacked a clear and well defined project scope. This could make respondents unclear about whether they satisfied the project scope.

The last question in this section considered project schedule management. This was presented as:

CS_32: A CubeSat project schedule was well managed.

Table 5.34. Managing project schedule.

	Frequency
Agree	77
Disagree	31
Neither agree nor disagree	8
Strongly agree	2
Strongly disagree	4
Frequency Missing = 5	

Table 5.34 presents data on whether CubeSats' project schedules were well-managed or not. A majority (60.63%) of respondents indicated that they were, while 24.41% expressed disagreement. This suggests that most CubeSat projects had a defined schedule, and successfully delivered the satellite on time. Development of a project schedule and strict adherence to it are essential elements of a project, and teams effectively manage their time using such strategies.

Tables 5.21–5.34, present frequency data with the average Likert-scale scores of survey respondents. The data presented in the frequency tables is based on an ordinal scale, which can not be tested for normality. However, it is still possible to apply the normality test by converting the ordinal measures into metric data. This is done by summing all scores from CS_19 to CS_32 into one metric data and applying the normality test. Table 5.35 shows a descriptive Likert-scale with its statistical measures, while Figure 5.18 presents a histogram graph that shows the frequency distribution of data.

Table 5.35. Basic statistical measures.

Location		Variability	
Mean	46.74590	Std Deviation	5.05435
Median	47.00000	Variance	25.54647
Mode	48.00000	Range	33.00000
		Interquartile Range	6.00000

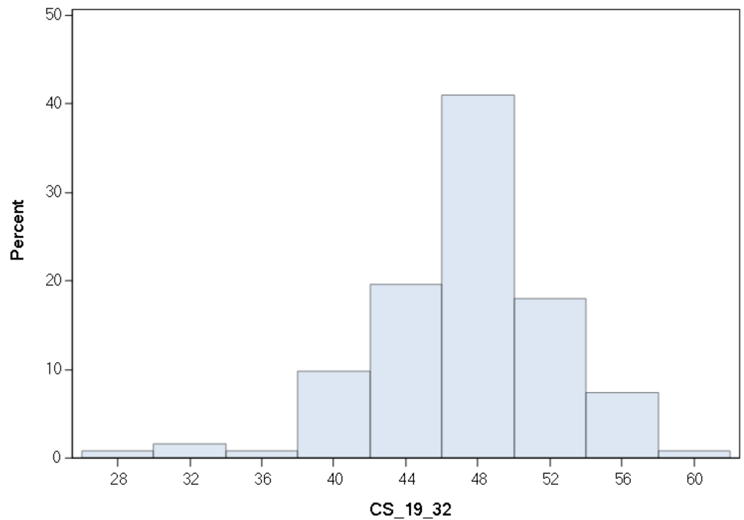


Figure 5.18. Distribution of data frequency of Likert-scale scores.

By summing all Likert-scale data and performing the normality test, it is shown that the data is relatively normally distributed.

In summary, Chapter 5 has presented, analyzed and discussed the results of statistical tests applied to the survey data. Section 5.1 presented the analysis and coding processes for the collected textual data (open-ended questions) from the 127 respondents using Windows NVivo12. The data was then grouped together for thematic content analysis to identify the words most frequently used by the participants. Section 5.2 showed the statistical testing results for all hypotheses. The results showed how factors affect CubeSat missions' success rate. In section 5.3, cross-tabulation analysis was presented to show the association between frequency tables and the correlation between them. Finally, section 5.4 presented tables with the applied statistical analysis for all Likert-scale statements.

6. DISCUSSION[6]

The value of CubeSat systems is significant (see section 1.4). They have gained popularity in educational institutions and for commercial use. CubeSats have attracted educators and manufacturers due to their ability to be quickly produced, their low cost, small size and low mass levels. On the other hand, CubeSats that are developed by aerospace industry firms can have higher cost levels, longer development time, and typically are more reliable [1]. While developers can swiftly design and build their CubeSats with a team of students from different disciplines using COTS parts, this does not guarantee that the CubeSat mission will be successful, as discussed in this dissertation.

6.1. CubeSat Project Lifecycle

CubeSat design and development have been mostly done without following typical engineering processes [72]. These practices may lead to CubeSats' unreliability. The proposed approaches are intended to facilitate systematic design and development and to address CubeSat reliability, traceability, and reusability issues. These approaches use a process methodology that focuses on translating mission objectives into basic building blocks, components, and tasks [80-89]. Additionally, they facilitate a bottom-up development model that uses defined engineering steps and processes.

This dissertation has identified a set of key factors, identified independent variables and measured their effectiveness on CubeSat systems. It seeks to increase CubeSat missions' success rates by developing systems engineering methodologies and tools. It evaluates the benefits of applying systems engineering methodologies and practices, which can be applied at different stages of the CubeSat project lifecycle and across different CubeSat missions.

Several challenges were discussed in the previous chapters that are reasons for CubeSat mission failure. These include the Inter-Integrated Circuit (I2C) data bus causing satellite failures, bus lockups, and power distribution failures (if the power lines are not overcurrent protected).

[6]Based on A. Alanazi and J. Straub. 2020. Defining CubeSat Mission Success and Assessing Reasons for Mission Failure. Under preparation for submission to the International Journal of Aerospace Engineering. and A. Alanazi and J. Straub. 2020. Evaluation of Software Engineering Practices Application to CubeSat System Development. Under preparation for submission to Acta Astronautica.

Previous studies have presented limited data regarding CubeSat mission failure reasons. This study suggests that tracking the reasons for these failures can help developers build more reliable and robust CubeSats. While several researchers track successful CubeSats (e.g.,[1][5]), there are a limited number of researchers (e.g.,[18][66]) that track failures. More work on tracking the issues responsible for mission failure and investigating the causes of these failures is needed. This dissertation identifies some critical factors that lead to CubeSat mission success. This was done by evaluating hypotheses to measure systems engineering decisions' impact on CubeSat systems' performance.

Since their introduction in 1999, over 1200 CubeSats have been launched (as of the end of 2019) [6]. Of these, about 400 "university-class" CubeSats were not successful [6]. This high rate of failure suggests that analysis is needed as to the causes of failure. This analysis demonstrated the need for a refinement of development process and for further analysis to facilitate the understanding and identification of factors that may lead to possible failures, before launch.

The research methodology that was used began with identifying six factors associated with CubeSat mission success. Based on these, a research question and six research hypotheses were developed. After analyzing these factors (see section 5.2), results showed that all six factors, and some sub-factors, all significantly affected CubeSat missions' success.

Defining success is important for every project, since having a defined goal is key to a team's ability to complete the project within the required time and budget. The survey responses had definitions of success that were divergent, based on respondents' expertise levels and the profession of the surveyed individuals. Those who had limited experience in developing CubeSats defined success as completing a mission within the scheduled time and budget. However, those with more experience defined success as achieving overall mission objectives, from planning to launching the CubeSat and receiving data.

In the survey responses from individuals that have built and used CubeSats, several factors were identified as crucial for CubeSat missions' success. These factors included identifying a purpose and vision for the mission. Results also suggested the importance of establishing a program structure for CubeSat development. Different organizations had different team compositions, systems engineering practices, and review approaches. In organizations that built CubeSats for academic purposes, the teams had higher turnover rates, due to student graduation. Mentors generally come from industry. Typically, the more experienced the mentor, the higher the mission success rate, and vice versa (see Section 2.3).

Risk analysis is also considered in this research. CubeSat developers have to select what risks they devote resources to. This can be done through cost to risk-reduction ratio analysis. Developers should also consider the trade-off between the potential for increased cost and delayed schedule versus the risk of on-orbit failure. Survey results indicated the importance of analysis and design. For CubeSats, it is essential to design for simplicity and robustness, as simple designs have fewer failure modes and are more likely to be achieved within typical CubeSat mission timelines.

Finally, the importance of testing was identified. Organizations building and launching CubeSats typically used full-system functional testing [72]. This includes end-to-end functional testing. The tests implemented by most organizations included a command execution test, a day-in-the-life test, an end-to-end communication test, and a complete power system charge/discharge cycle. Other factors included identifying common CubeSat failures, part quality issues, documentation, and ensuring the launch schedule does not compromise the development process (see Section 5.2).

This research also sought to identify the most challenging activities in the CubeSat project lifecycle. It was notable that many project challenges occur due to software and hardware issues (see Section 5.1). This highlights the importance of the design, testing, and integration of CubeSat components. Another issue mentioned was power. In some cases, there is not enough surface area on a CubeSat for sufficient power generation. Deployable arrays, which are a solution to this, add significant complexity to satellite design. Another issue is late decisions made on selecting tools for a project. The software used in some projects also presented challenges. These range from flight software design being insufficiently considered, to having no debugging support, to integration and testing processes issues.

6.2. Hypotheses

Six hypotheses were developed to evaluate the benefits of applying systems engineering methodologies and how effective they can be in reducing CubeSats' mission failure rate. Each was tested using statistical analysis of responses to several survey questions.

The first tested hypothesis (H1), defining mission objectives (DMO), assesses the importance of this mission planning step. Strong objectives can guide developers during the entire project and act as a point of reference, hence increasing the project's chance of success. DMO was assessed based on six questions that cut across the defining mission, perception of development status, and system requirements areas. Responses indicated that the developer's ability to identify the project's scope and goals at the early stages of development had a significant influence on the project's success.

The second hypothesis (H2) tested was whether timeline analysis (TA) positively affects CubeSat mission success. TA efficacy evaluation was based on sub-factors that address its effectiveness for CubeSat missions. These include questions regarding the development phase of the system, the type of the developed CubeSat, aspects of software quality, and factors associated with the CubeSat project's failure or success. Responses showed that implementing TA in a CubeSat project increases the CubeSat mission's likelihood of success.

The third hypothesis (H3) sought to measure whether mission-assurance analysis (MA) affects CubeSat mission success. MA analysis included sub-factors such as project challenges, reduced testing time, redesign scenarios, methodology type and selection, and stakeholders' engagement in the project.

The fourth hypothesis (H4) tested whether performing a critical design review (CDR) in the design phase has a positive effect on the mission's success rate. CDR efficacy was assessed based on sub-factors including project challenges, tool and model failures, reduced testing time, redesign scenarios, team size, and the most and least important aspects of software quality. Results showed that performing a critical design review (CDR) during the design phase positively affects a mission's success rate.

The fifth hypothesis (H5) tested whether experience in quality attribute (EQA) practices has a positive effect on a mission's success rate. Whether a correlation exists between missions' success, having expert personnel on the team, the number of years a developer has spent in this environment, and the number of CubeSat projects a developer has been involved with was also evaluated. Results indicated that experience in quality attribute practices has a positive effect on a mission's success rate.

Finally, the sixth hypothesis (H6) tested whether using a schematic diagram (SD) in a CubeSat project positively affects the mission's success rate. Sub-factors were also tested for SD efficacy assessment. These included the time needed for development, the methodology type selected, stakeholders' participation of the CubeSat development processes, and tool or model failures. Results indicated that using a SD in a CubeSat project positively affects the mission's success rate. SDs provide a means of visualizing system structure and operations. A SD helps developers to understand the system well enough to construct a complex system such as a CubeSat.

These factors have been used as systems engineering practices and show success in various sectors, including aerospace [26][67][80]. A developer's ability to determine the factor most associated with CubeSat project failure is essential for mission success. A CubeSat mission can have issues that inhibit its success. These issues can include mismatches between the expectations of the stakeholders and the team, and the resources available to the developer. In other instances, the developer may have been overly optimistic about what could be accomplished using the available resources [67]. Identification of key issues that can lead to failure facilitated their mitigate enhance of mission success.

6.3. Recommendations

Based on the results from this research, recommendations to CubeSat developers, that may help with their future projects, are now provided. These recommendations start with general recommendations. Then, processes useful during the first phase of the CubeSat project lifecycle, and those which have utility through the rest of the development stages are discussed.

A CubeSat should be designed for easy assembly and disassembly. Developers should seek to build an experienced team (see Section 5.1). A team should have experienced veterans and frequent informal peer reviews with proven subject experts. Developers should stock spare components. These spare components will support parallel hardware and software development and can serve as flight spares. Developers should perform systematic tests. Tests such as mission assurance analysis and timeline analysis are crucial. Developers should also maintain understanding of the costs involved for key components and how costs may change during the project.

At product initiation, it is important to determine the goals and project scope for the team. This means defining the mission objectives and defining what success at the end of the project is defined by. Small changes at the last stages of development could be very costly and lead to the project's failure. This justifies developing a full plan for the mission's time and budget (see Section 5.1, [14],[67]).

In this study, 24.41% of respondents indicated that project scope and goals were not clearly identified at the beginning of their projects, and they indicated that this could be a reason for their

CubeSat's failure. This indicates that respondents were aware of the need for identifying scope and goals prior to the design and testing phases of the project. Additionally, the developer's ability to identify the project's scope and goals at the early stages of development had a significant influence on the success of the project.

One respondent noted that "each one is unique; you must define your objectives from day one. Some missions are successful if they make it on time to the launch pad, others need to generate a significant amount of revenue. It's the goal that you define on day one and revise/review along the way that defines your success." Defining mission objectives is, thus, essential to CubeSat development teams. Team members may work in separate groups and have different backgrounds and disciplines. Having a clearly defined scope, short-and long-term objectives, and shared terminology can help to avoid misunderstandings between team members. These will also help enhance the interaction between CubeSat developers and stakeholders who come from different organizational levels and areas. Having well defined mission objectives at early stages of the development process can help avoid costly failures that might occur due to changes at later stages of the development process [14].

It is recommended to apply Timeline Analysis (TA) during CubeSat projects to help clarify the relationships between functions and tasks, and to identify the specific time allocated for design requirements. TA can show the required time needed, as well as the design constraints in place while building CubeSat systems. Respondents who were industry professionals indicated that they recognized the effectiveness of using TA in their projects and saw how TA could faciliate mission success. Some CubeSat developers may try to avoid TA due to concerns over budget and time. However, by doing so, they may end up with overlap between tasks and functions, and other issues. All of these can be reasons for mission failure (see Section 5.3).

Of these surveyed, 17% of respondents either did not know if the CubeSat they have designed had implemented techniques to reduce testing time or not. This may be because they have not reached the testing phase yet, because they lack something to compare their mission to, or because the respondent is not involved in the testing process. However, 60% of respondents indicated that their design had not reduced the testing time needed. Additionally, 20% of respondents indicated that it may take 3 to 4 years to develop a CubeSat because of schedule delays and shortage of COTS components.

TA is most typically used for products with longer schedules, larger budgets, and larger team sizes, due to the time cost of the TA process. It also requires experience in forecasting the allocated time for testing.

Running MA tests is important to ensuring missions' success [26][27]. It is recommended to apply MA analysis to spot any flaws or issues early in the development stage.

Some respondents suggested that following rapid development or Agile methodologies for their project would be desirable, since they have limited time and a limited budget for their project. Others, indicated that they preferred following different testing procedures. Some respondents thought that some needed testing for their project was ignored, such as, for example, thermal testing. Respondents also indicated that they have experienced challenges with testing procedures when adding or removing features or components after the design phase of the project.

MA tests include communication link testing, power system testing, thermal & vacuum testing (TVAC), and regular reviews (formal and informal). A standardized approach for MA analysis should be applied throughout the development processes. Although it is beneficial to use MA, many developers of CubeSats in educational institutions ignore this, due to the time required for MA activities [1][7].

Analysis indicated that having a critical design review (CDR) is associated with the success of a CubeSat mission. It is recommended to use a CDR and preliminary design review (PDR) in the development process. They can enhance documentation procedures, as well as enhancing the reusability, testability, and modifiability of the system. Survey results indicated that 17.32% of respondents found mission requirements complex or hard to follow due to a lack of proper documentation. Respondents also indicated that project specifications contained non-testable functional requirements. The fact that respondents indicated the presence of non-testable functional requirements means that testing could not have been fully effective. The project team should be confident of requirements and their testability. Deviation from this may result in wasted time, effort and other resources during the development process.

In some cases, a CDR is not cost effective, despite the benefits that it could provide. A PDR is needed, at least, for simple missions for educational purposes. Having both a PDR and CDR is recommended for larger projects and complex missions.

Developers' experience with quality attribute practices, including usability, security, and reliability, can help increase the success rate of a CubeSat mission. Survey results show that projects with experts are typically more successful than projects lacking experienced individuals. The number of years a developer spent in CubeSat or related development, and the number of CubeSat projects a developer has been involved with correlate with having a successful project. It is recommended to have at least one expert member on the team with experience in building CubeSats. EQA deficiency was not an issue identified by industry professionals. A lack of EQA was mostly identified for projects in educational institutions. If an expert is not available, developers are recommended to attend workshops on building CubeSats. Students might also seek an internship at a company that manufactures CubeSats [7][58]. Of those surveyed, 66% of total respondents indicated that their CubeSat projects were for educational purposes. Of those responding, 46.45% indicated that their project had difficulty with complex requirements and that the requirements were hard to follow. These results suggest that respondents may see requirements as a big issue in CubeSat development, either due to ambiguity in them or, perhaps, due to a lack of experience.

Finally, it is recommended to use a SD for CubeSat projects to achieve the visibility of components and the traceability of requirements. SDs can depict hardware and software components and their interrelationships. Of those surveyed, 37% of respondents indicated that CubeSat system failures occur due to the failure of models. Respondents indicated that there are no standardized models used in the design phase of the project. Thus, a SD is recommended to be used for a project because it can resolve these issues and illustrate the interfaces between system components and the spacecraft's functional architecture. A detailed SD can be used to track requirements and functional activities. Using a SD can help to spot any issues or defects in earlier stages of development.

Other recommendations that help ensure CubeSat missions' success include defining the scope, goals, and success criteria at the program start. This definition justifies the time, budget, and resources available for the program. It is also recommended to conduct risk-based mission assurance. Performing a risk assessment at the start of the program and repeatedly reviewing it enables the prioritization of analyses, tests, reviews, and activities. Developers should also plan for sufficient integration, verification and validation and develop a timeline. Developers should design for simplicity and robustness. This is done by assuming that designs will fail and proving that they will work (see Section 5.3, and [2]).

Analyzing all of these factors can help CubeSat developers to design better projects. These factors can be transformed into critical systems engineering processes that can help in the design and operations of CubeSats. The best way to facilitate this transformation is to design systematic frameworks that can decrease problems and issues when developing CubeSats (see Section 1.5). CubeSat frameworks should provide a clear vision and well-defined system requirements, that are interpreted from stakeholder needs. They also should provide CubeSat developers with time to implement the best systems engineering practices and processes such as verification and validation methodologies, documentation, and requirements elicitation.

A number of different methodologies and approaches can be applied to a CubeSat project. However, modeling and frameworks designed explicitly for CubeSats are minimal. At its most basic a CubeSat framework should include a clear vision of the project scope, scheduling, and budget to avoid problems and issues that may occur at a later stage of development. It should also include timeline analysis to help in allocating testing and other time during the project. Additionally, the framework can include other mission-assurance analysis to help developers to complete the project without any problems or issues. Development best practices include well-defined testing procedures and the use of regular reviews. Best practices also include a preliminary design review (PDR) and a critical design review (CDR), the identification of non-functional requirements and documentation processes. Goals of this development process include modifiability, simplicity, and reusability.

6.4. CubeSat Preliminary Model

In Chapter 5, respondents indicated that many CubeSats fail due to model issues rather than tool issues. They also indicated that miscalculations and incorrect hardware assembly are reasons of CubeSat failure. Respondents also pointed out that adding new features and changing requirements will also lead to CubeSat failure. Further, they indicated that project challenges mostly occur in planning, budget, and scheduling. Additionally, respondents indicated that other failure issues were due to issues with deployable structures' design, failure mode effects and criticality analysis (FMECA), fault detection, isolation, and recovery (FDIR), and the complexity of flight software systems architecture. Thus, results and analysis from Chapter 5 show that system architectural models and frameworks can help in increasing the CubeSat mission success rates.

This section includes an example of a preliminary Cube Satellite model using AADL. This preliminary design contains a graphical representation of a CubeSat's subsystems. It uses the AADL modeling language to provide a logical architectural view of hardware and software components and their interaction. This model is an example of an educational CubeSat. Principles of AADL and other modeling frameworks that, related to space environment, were discussed in (section 2.3). Previous studies [80-89] that covered the fundamental challenges of non-functional requirements (NFRs) including, security, usability, safety, and deployabliliy, for CubeSats served as a basis for developing the AADL model presented in this section. Other studies [24] covered the architectural systems design that is required for requirements elicitation and the refinement of end-to-end quality attributes.

This AADL model focuses on the CubeSat's nonfunctional requirements and not data structures or algorithms. It shows the CubeSat's subsystems' structure, including the ADCS, EPS, OBC, radio, communication, and sensors subsystems. These subsystems represent the standard core components of a typical 1U CubeSat (see section 1.2 and [7],[15-19]). This model does not include specific values or system inputs and parameters such as the mission requirements, mass budget, and orbital parameters which would be required for specific missions.

This model was created using AADL [101] and developed using OSATE2 [79], an eclipse-based tool for modeling in AADL. The main types of components that are presented in this model are software, hardware, and composite. The software is a process or a thread where the program is executed. The hardware is memory where the program is stored. Composite includes the internal structures of subcomponents. The model also includes features to show the mechanisms of communications between components. The platform components, depicted in Figure 6.1, include shapes that represent device interfaces to external environments, systems (such as the OBC system), busses (which provide physical connectivity between hardware), processes (such as an OBC controller), ram for code storage, and threads (such as reset systems, and data conversion).

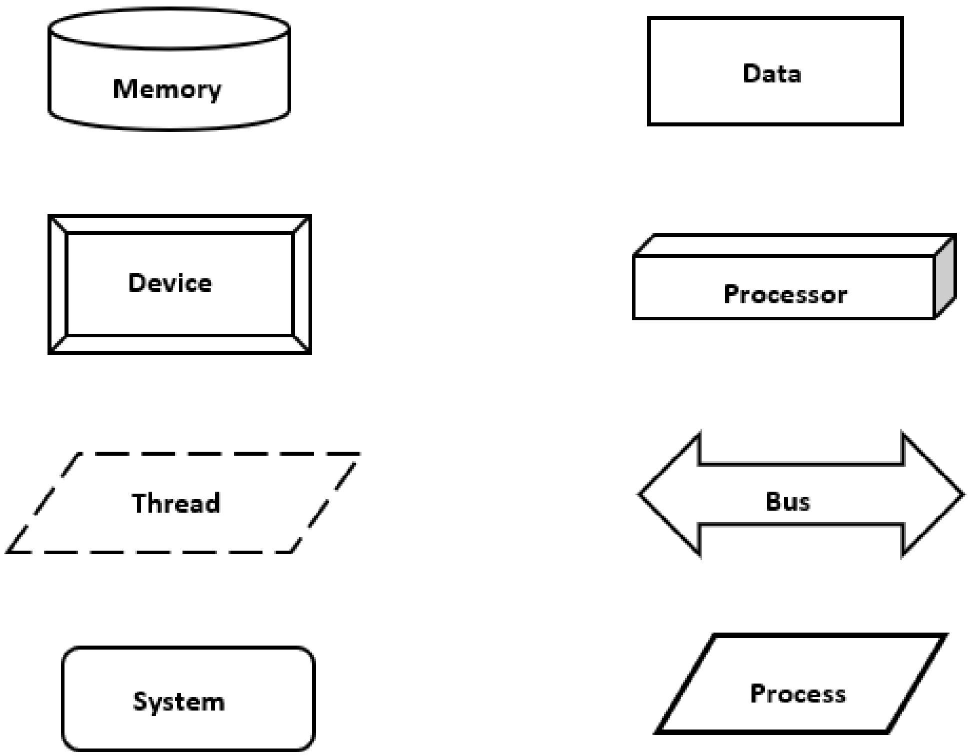


Figure 6.1. AADL graphical notion.

An example of the OBC system components and subcomponents, shown in Figure 6.2, illustrates how the components are defined and how the internal structure of a device is defined inside an implementation. The AADL construct is a mechanism to model components. It provides a useful abstraction of entities that contain complex hardware, such as an analog to digital converter (ADC), and associated software, allowing data communications to be modeled. The example OBC CubeSat subsystem contains multiple devices including a WDT, which is used to detect and recover from computer malfunctions, and an ADC, for analog to digital conversion.

```
package OBC
public
   system OBCSys

   end OBCSys;

   system implementation OBCSys.obc
       subcomponents
           this_MCU: processor MCU;
           this_codeStorage: memory codeStorage;
           this_ram: memory ram;
           this_bulkMemory: memory bulkMemory;
           this_ADC: device ADC;
           this_WDT: device WDT;
           this_HWConnect: bus HWConnection.obc;
           this_process: process OBCController.obc;

       connections
           bus_processor: bus access this_HWConnect -> this_MCU.bus_access;
   end OBCSys.obc;
```

Figure 6.2. Defining systems components and subcomponents for the OBC system.

The model includes features to show the communications between components. Features can specify port names and types, as depicted in Figure 6.3.

```
device speed_sensor
    features
        bus_access: requires bus access OBC::HWConnection;
end speed_sensor;

device implementation speed_sensor.adcs

end speed_sensor.adcs;

device reaction_wheel
    features
        bus_access: requires bus access OBC::HWConnection;
end reaction_wheel;

device implementation reaction_wheel.adcs

end reaction_wheel.adcs;
```

Figure 6.3. Ports and features of the ADCS system.

AADL uses features to specify the port name and the type of port used for communications among components. It uses ports for communication interfaces to exchange data or events. Ports need to be defined for each device inside its component type definition. In the ADCS, two devices are declared: speed_sensor and reaction_wheel. Both have an output data port, bus_access, as shown in Figure 6.3. Additionally, the reaction_wheel device models the reaction wheel, which enables accurate control of the attitude of a CubeSat. The speed_timer device models a speed timer which provides a high-resolution measurement of the time interval needed to charge the capacitor. It is important to note that the data types in this example are not declared. This is done to show that untyped data declarations are supported. Models for both devices (speed_timer and reaction_wheel) might not have specific data types during the early stages of modeling.

The CubeSat mission success factors discussed in Section 5.2 were used as the basis for building this preliminary AADL CubeSat model. AADL modeling may help CubeSat developers in resolving the difficulties associated with their limited payload space, OBC, and communication system capabilities. It may also help in increasing reliability, where risk comes from launching CubeSats before they have been extensively tested and other factors.

A CubeSat project, as discussed in Section 1.4, should include methodologies, testing scenarios, communications, documentation, and timeline analysis, as engineering steps. This AADL model can provide several benefits in these areas, when used for a CubeSat project. These include increased project control, earlier reduction of project risk, improved efficiency, and enhanced analysis of stakeholder needs.

The benefits of using an AADL model for CubeSat development relate to the previously discussed concepts of systems engineering process, steps and practices. Models provide benefits in the design and management of complex systems, such as CubeSats. Using a model can help designers and implement understand how pieces of the spacecraft fit together into a consistent whole. Models can also be used to capture functions, behaviors, structures, flow, interfaces and ports. A well-designed model can be helpful to identify and fix some issues that occur during the CubeSat development process (see Section 5.2). A model also can be used as a reference for reducing risk through the analysis of mission, timeline, and quality assurance. This will enhance the reliability and availability of CubeSat systems (see Section 3.1).

The benefit of modeling a CubeSat system using AADL is principally being able to capture systems requirements for the CubeSat. Requirements elicitation was one of the major issues identified as causing CubeSat mission failure (see Section 5.3). The first step in developing any complex software system is creating models (for behavior, data, functional characteristics and other areas) before beginning implementation [79]. By modeling the system, problems and issues that could lead to a CubeSat mission can be better understood. In Sections 5.1 to 5.3, several issues were identified by survey respondents, including the ambiguity of requirements, that can be solved by modeling. Thus, a CubeSat project designer would benefit from using an AADL model to verify operations efficacy and quality during the design, development, and integration of the system. Through modeling, it is possible to spot errors at early stages of the development of a CubeSat. Preventing problems can lead to a reduction in the time, cost, and effort required for development.

7. CONCLUSION

CubeSat systems have gained popularity for many educational and commercial uses. They have attracted manufacturers and educators because of their low cost and capabilities. While many missions do succeed, the use of a team of students with COTS parts does not guarantee that a CubeSat will be successful. Statistics have shown that CubeSat missions' average failure rate is higher in academic and research institutions than in commercial or government organizations [5][6]. Reasons for the higher failure rate may include mechanical, communications, and system design issues. Many problems are due to the process of designing and developing the CubeSat. To increase CubeSat missions' success rate, proper systems engineering steps and processes need to be implemented during the development cycle. Using these recommended processes can also help CubeSat designs and systems to be more secure, reusable, and modular.

The research presented herein has identified independent variables related to CubeSat mission success and measured their effectiveness. It seeks to help increase the CubeSat mission success rate by developing systems engineering methodologies and tools. It has also evaluated the benefits of applying systems engineering methodologies and practices. These can be applied at different stages of a CubeSat project's lifecycle and used across multiple CubeSat missions.

CubeSats are used to carry out various space missions for commercial, scientific, government, and military purposes [5]. They are regularly used for Earth observations [35] and amateur radio [19]. Most CubeSats are designed for and intended to be put in low-Earth orbits (LEO) with an altitude between 160-2,000 km above the Earth's surface; however, some CubeSats are now being built for deep-space applications [34].

Data [5] shows that, since 2014, the use of CubeSats to perform industrial work and commercial projects has increased. The number of launched CubeSats has dramatically increased in the last eight years [6]. As of January 2020, more than 1,200 CubeSats have been launched [6]. The wide use of CubeSats has been enabled by standardization [67] and the use of a common CubeSat launch dispenser. The dispenser ensures that the CubeSat and the launch vehicle are well protected from each other [14].

The development of a CubeSat system has become easier for educational institutions, industry, and even hobbyists in recent years. CubeSats are designed using different configurations depending on their intended mission. They have common subsystems components such as the antennae, batteries, sensors, an OBC, and a radio transmitter used for downlink and uplink.

The system engineering process seeks to ensure that the complex CubeSat system operates successfully. It ensures that designers understand the requirements of the system. Several different systems engineering methodologies and approaches can be applied to a CubeSat project. CubeSats have high demand because they are less expensive, more flexible, and developed faster. This dictates a need for different systems engineering processes as opposed to larger spacecraft. The adoption of a systems engineering methodology helps reduce the rate of CubeSat mission failure. The different hypotheses that were evaluated in this work focused on the reasons behind CubeSat missions' failure. This dissertation proposes particular practices to be applied in the CubeSat development environment to enhance mission success.

These enhanced practices are needed as more than 40% of CubeSats failed to meet their expected mission objectives [5]. The failures were, in some cases, attributed to fast orbit decay and reentry (because they are launched into LEO). Also, some of the practices that were applied by the students were considered to be of low quality and caused failures. In other cases, components that were used in the development process were not up to the required standards, leading to failure. Approaches to enhance CubeSat mission success were developed, based on best practices. For instance, some educational institutions used the concepts of guidance and specifications to help developers achieve mission success [50][51][52]. This approach ensured that students got fully engaged in the project and drove mission success. Other institutions applied quality assurance practices to facilitate the success of projects. They also used different methods such as, requirements-mapping matrix, and tools, including software, to enhance mission success [23][64][65].

This dissertation used a hybrid method of research and a mixture of qualitative and quantitative questions and analysis. From this, insights have been provided which help in understanding the reasons for frequent failure among CubeSat missions. Quantitative methods have been used to enable quantification of the collected qualitative data. Through this, this study has identified a gap in knowledge calling for further research. It has done this by evaluating six independent variables associated with CubeSat mission success. By evaluating these factors, the research explains how variables correlate with mission success. They were used to develop the model presented herein. To increase the likelihood of mission success, it is essential to define the scope of the mission and its success criteria. From these, it is then essential to define mission objectives.

Organizations should pay particular attention to the type of CubeSat mission they are developing, since success criteria typically vary by the type of mission. For instance, the criterion for students may be completing their project within time and budget targets. In contrast, for an industry mission, a successful mission could be expected to survive and successfully operate for a 60-day or more period on orbit. CubeSat development teams may never discuss the scope and objectives of their mission. While this is not ideal for mission success, it may not always lead to mission failure.

It is essential to allocate sufficient testing time for each stage of the project life cycle. Choosing the proper testing methods and allocating an adequate amount of time for each test is also critical. Notably, the time needed for testing tends to vary as a result of different levels of mission complexity. Design choices can increase or reduce the amount of testing required. This work, thus, sought to determine whether the CubeSat design used reduced the testing time needed. A relationship between the CubeSat functions and required testing tasks was identified and included in a proposed timeline analysis technique, which is a critical systems engineering process for a CubeSat project.

Timeline analysis is essential in showing the needed time, given the design constraints in place. It can help define how much time is needed for each task. This will help developers when they are planning the schedule to complete their tasks. It is also important in prioritizing tasks and deciding which ones are assigned more time than others. Time is an essential consideration in CubeSat systems development because hardware and software systems must be tested within a timeframe that may be constrained by launch provider and other deadlines. This raises the need to define the types of tests, and identify the required facilities in the laboratory. The laboratory facilities must be assessed to determine if they can perform the tests according to the specifications.

Survey responses indicate variation based on the expertise and the profession of the respondent. Individuals with limited experience in developing CubeSats defined success as the development being completed within the scheduled time and budget, while individuals with more experience in developing CubeSats defined success as achieving overall mission objectives, including launching the CubeSat and receiving data.

Most respondents indicated projects experienced both hardware and software issues. Respondents also indicated a consistent and serious issue with power. There is not enough surface area on most CubeSats for sufficient power generation, and deployable arrays add significant complexity to satellite designs. Respondents also identified that it is essential to decide on the tools that will be used early in the process as opposed to deferring this decision to a later phase of development. Respondents also discussed that it is beneficial to incorporate only the necessary functionality during the design phase.

Through testing of the six hypotheses that are the primary analysis in this dissertation, it has been determined that the identified factors tend to affect a CubeSat mission's success rate. Each of the factors has been defined along with multiple sub-factors which were presented as survey questions. These hypotheses and factors were used to identify proposed changes to the systems engineering process and practices used during the CubeSat development life cycle.

To help CubeSat developers succeed in their future projects, this research suggests that they begin by setting up goals and project scope for the team at the earlier stage of development possible. They should define the objectives of their mission and the definition of mission success If changes occur during later stages of development, they can be very costly, and might result in the project's ultimate failure.

The benefits of defining mission objectives (DMO) were measured by measuring the number of CubeSat missions that have succeeded and whitch clearly defined their mission at an early stage in the project's life cycle. The study also investigated how CubeSat missions succeeded without defining their mission objectives.

The second factor identified was timeline analysis (TA). Due to mission complexity, the timeline may vary; hence, it is crucial to determine if CubeSat design changes can help in reduce the required testing time.

In this study, the impact of design on testing requirements was assessed and correlation between complexity and testing time was shown. The third factor considered was mission assurance (MA). Mission success requires running risk management processes. Engaging in risk management helps to identify flaws and issues at early stages of CubeSat development.

The fourth factor identified was the use of a critical design review (CDR). System design complexity is common in CubeSat development and simple design is essential for mission success. Thus, it is essential to use best practices and to have team members with experience with design problems that are likely to be encounter during development.

The fifth factor was having team members with experience in quality attributes (EQA). This factor studied the challenges that developers have encountered and their resolutions. To decrease the failure rate and increase the CubeSat system success rate, CubeSats need to be more secure, modular, and reusable. This factor also measured developers' experience with quality attribute practices including usability, security, and reliability.

The last factor considered was the use of schematic diagrams (SD). To have a robust CubeSat system, developers typically must use at least some commercial-off-the-shelf (COTS) parts. Some missions will also require parts that are radiation hardened and can survive in a space environment. SDs can help define the interactions between these various parts.

Results from this research should be particularly valuable to CubeSat developers who are new to satellite development. This research focused on investigating whether systems engineering processes and practices (DMO, TA, MA, CDR, EQA, and SD) positively affect the CubeSat mission success rates. It was found that they did, several assumptions regarding CubeSat developers' technical challenges and experience levels were also confirmed.

This study also found that most CubeSat developers are open to the modification of requirements during the testing phase. Respondents also indicated that a smaller team size of CubeSat developers positively impacts a project, as a greater number of team members increases miscommunications between members and can lead to schedule delays. Requirements were also identified as a big issue in CubeSat development. Some requirements are hard to follow, due to ambiguity and developers' lack of experience. Additionally, some functional requirements must be modified to be testable. The project team must be confident in their functional requirements and their testability. Issues with requirements may result in time, effort and resources being spent trying to develop and test the wrong things. Applying appropriate methodologies in CubeSat projects will have a positive effect on the CubeSat mission's success. The exclusion of a needed process or methodology may cause an inability to achieve the core objectives of the project.

This dissertation also considered the use of AADL for CubeSat modeling. AADL provides a graphical representation of a CubeSat's subsystems. It provides a logical architectural view of the hardware and software components and their interactions. A model was discussed which is an example of an educational-style CubeSat. It does not include specific values or system inputs and parameters (such as mission requirements, mass budget, and orbital parameters). Thus, future work will involve in designing an engineering tool based on the identified factors that includes more detail subsystems including inputs and parameters.

In conclusion, using the recommended processes discussed in this research would be beneficial to CubeSat developers. This research has been able to identify independent variables and measure their effectiveness on CubeSat systems. It is imperative to institute engineering methodologies and processes in the design and development of CubeSat projects. This research has determined that the success of a CubeSat mission is positively influenced by applying systems engineering processes.

REFERENCES

[1] M. Swartwout, and C. Jayne, ”University-Class Spacecraft by the Numbers: Success, Failure, Debris. (But Mostly Success)”, in the 30th Annual AIAA/USU Conference on Small Satellites, Logan, UT, 6-11 August 2016, paper SSC16-XIII-1.

[2] M. Langer, A. Lill, F. Schummer, M. Weisgerber, S. Rückerl, and A. Hoehn, “Reliability Assessment and Reliability Prediction of CubeSats through System Level Testing and Reliability Growth Modelling,” in the 69th International Astronautical Congress, Bremen, Germany, Oct. 2018.

[3] A. Chin, R. Coelho, R. Nugent, R. Munakata, and J. Puig-Suari, “CubeSat: The Pico-Satellite Standard for Research and Education,” in AIAA SPACE 2008 Conference & Exposition, 2008, no. February 2015, doi: 10.2514/6.2008-7734.

[4] J. D. Moses, D. L. Pierce, M. S. Seablom, and A. J. Petro, “An Overview Of The NASA/ Science Mission Directorate CubeSat Activities,” AIAA/USU Conf. Small Satell., 2016.

[5] M. Swartwout, “Reliving 24 Years in the Next 12 Minutes: A Statistical and Personal History of University-Class Satellites,” 2018, [Online]. Available: https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=4277&context=smallsat.

[6] E. Kulu, “NanoSat_DB.” https://www.nanosats.eu/ (accessed Jan. 15, 2020).

[7] NASA CubeSat Launch Initiative, “CubeSat 101: Basic Concepts and Processes for First-Time CubeSat Developers,” no. October, p. 96, 2017, [Online]. Available: https://www.nasa.gov/sites/default/files/atoms/files/nasa_csli_cubesat_101_508.pdf.

[8] D. Kaslow, B. Ayres, P. T. Cahill, L. Hart, and R. Yntema, “Developing a CubeSat Model-Based System Engineering (MBSE) reference model - Interim status #3,” 2017, doi: 10.1109/AERO.2017.7943691.

[9] F. C. C. (FCC), “Regulatory Agency.” https://www.fcc.gov/licensing (accessed Jan. 15, 2020).

[10] NASA, “Regulatory Agency.” https://public.ksc.nasa.gov/lspeducation (accessed Jan. 15, 2020).

[11] N. O. and A. Administration, “Regulatory Agency.” https://www.nesdis.noaa.gov/CRSRA/noaaLicensees.html (accessed Jan. 15, 2020).

[12] National Science Foundation (NSF), “Regulatory Agency.” https://www.nsf.gov/funding/index.jsp (accessed Jan.15, 2020).

[13] U. A. R. Lab, “Regulatory Agency.” https://afresearchlab.com (accessed Jan. 15, 2020).

[14] J. R. Wertz, D. F. Everett, and J. J. Puschell, Space mission engineering: the new SMAD, 3rd ed. Torrance: Microcosm Press, 2011.

[15] NASA, “State of the Art of Small Spacecraft Technology,” State Art Small Spacecr. Technol., no. December, pp. 1–202, 2018, [Online]. Available: https://sst-soa.arc.nasa.gov/04-propulsion.

[16] M. C. Stephen Waydo and D. Henry, “CubeSat Design for LEO-Based Earth Science Missions,” Compend. Res. Math Education, pp. 3–27, 2002.

[17] ISISSpace, “CubeSat Components,” [Online]. Available https://www.isispace.nl/products/ (accessed Jan. 15, 2020).

[18] A. Alanazi and J. Straub, “Engineering methodology for student-driven CubeSats,” Aerospace, vol. 6, no. 5, 2019, doi: 10.3390/AEROSPACE6050054.

[19] “ArduSat 1U configuration.” https://amsat-uk.org/2014/01/19/deploying-software-updates-to-ardusat-in-orbit/ (accessed Feb. 02, 2020).

[20] P. Muri and J. McNair, “A survey of communication sub-systems for intersatellite linked systems and cubesat missions,” J. Commun., vol. 7, no. 4, pp. 290–308, 2012, doi: 10.4304/jcm.7.4.290-308.

[21] D. L. Iverson, “System health monitoring for space mission operations,” IEEE Aerosp. Conf. Proc., no. June, 2008, doi: 10.1109/AERO.2008.4526646.

[22] A. Balador, A. Kouba, D. Cassioli, F. Foukalas, R. Severino, D. Stepanova, G. Agosta, J. Xie, L. Pomante, M. Mongelli, and P. Pierini, "Wireless communication technologies for safe cooperative cyber physical systems," Sensors, vol. 18, no. 11, pp. 1–27, 2018, doi: 10.3390/s18114075.

[23] "ESA." https://esmat.esa.int/ (accessed Dec. 05, 2019).

[24] M. Gagliardi, W. Wood, and T. Morrow, "Introduction to the Mission Thread Workshop," Softw. Eng. Inst., no. October, 2013, [Online]. Available: http://repository.cmu.edu/sei/762.

[25] A. Alanazi, "Methodology and tools for reducing cubesat mission failure," in 2018 AIAA SPACE and Astronautics Forum and Exposition, 2018, pp. 2–9, doi: 10.2514/6.2018-5122.

[26] Andrew P. Sage and William B. Rouse, Handbook of Systems Engineering and Management - Andrew P. Sage, William B. Rouse - Google Books. John Wiley & Sons, 2011.

[27] R. A. Austin, R. D. Schrimpf, A. F. Witulski, N. Mahadevan, G. Karsai, B. D. Sierawski, and R. A. Reed, "Capturing and Modeling Radiation Hardness Assurance throughout the Project Lifecycle," 2018. Accessed: Jul. 21, 2019. [Online]. Available: https://ntrs.nasa.gov/search.jsp?R=20190001591.

[28] A. Tierno, M. M. Santos, B. A. Arruda, and J. N. H. Da Rosa, "Open issues for the automotive software testing," 2016 12th IEEE Int. Conf. Ind. Appl. INDUSCON 2016, no. November, 2017, doi: 10.1109/INDUSCON.2016.7874609.

[29] N. Lee, Digital Da Vinci: Computers in the arts and sciences, no. January. 2014.

[30] M. Yaseen, N. Ibrahim, and A. Mustapha, "Requirements prioritization and using iteration model for successful implementation of requirements," Int. J. Adv. Comput. Sci. Appl., vol. 10, no. 1, pp. 121–127, 2019, doi: 10.14569/IJACSA.2019.0100115.

[31] A. Toorian, K. Diaz, and S. Lee, "The CubeSat approach to space access," IEEE Aerosp. Conf. Proc., vol. 1, no. 1, 2008, doi: 10.1109/AERO.2008.4526293.

[32] R. Rose, J. Dickinson, and A. Ridley, “CubeSats to NanoSats; bridging the gap between educational tools and science workhorses,” IEEE Aerosp. Conf. Proc., 2012, doi: 10.1109/AERO.2012.6187417.

[33] J. M. Gambi and M. L. García del Pino, “Autonomous shooting at middle size space debris objects from space-based APT laser systems,” Acta Astronaut., 2017, doi: 10.1016/j.actaastro.2016.11.026.

[34] I. Garrick-Bethell, R.P. Lin, H. Sanchez, B.A. Jaroux, M. Bester, P. Brown, D. Cosgrove, M.K. Dougherty, J.S. Halekas, and D. Hemingway, “Lunar magnetic field measurements with a CubeSat,” Sens. Syst. Space Appl, 2013, doi: 10.1117/12.2015666.

[35] D. Selva and D. Krejci, “A survey and assessment of the capabilities of Cubesats for Earth observation,” Acta Astronaut., vol. 74, pp. 50–68, 2012, doi: 10.1016/j.actaastro.2011.12.014.

[36] T. Baca, M. Platkevic, J. Jakubek, A. Inneman, V. Stehlikova, M. Urban, O. Nentvich, M. Blazek, R. McEntaffer, and V. Daniel, “Miniaturized X-ray telescope for VZLUSAT-1 nanosatellite with Timepix detector,” Journal of Instrumentation, vol. 11, no. 10, 2016, doi: 10.1088/1748-0221/11/10/C10007.

[37] “Open Source Initiative.” https://www.opensource.org (accessed Dec. 10, 2019).

[38] “Open Source Hardware Association.” https://www.oshwa.org (accessed Dec. 10, 2019).

[39] M. Swartwout, “CubeSat_DB.” https://sites.google.com/a/slu.edu/swartwout/home/cubesat-database (accessed Dec. 10, 2019).

[40] A. Macario-Rojas, K. L. Smith, N. H. Crisp, and P. C. E. Roberts, “Atmospheric interaction with nanosatellites from observed orbital decay,” Adv. Sp. Res., vol. 61, no. 12, pp. 2972–2982, 2018, doi: 10.1016/j.asr.2018.02.022.

[41] M. Langer and J. Boumeester, “Reliability of CubeSats – Statistical Data, Developers’ Beliefs and the Way Forward,” Proc. 30th Annu. AIAA/USU Conf. Small Satell., 2016.

[42] E. Deems, “Risk management of student-run small satellite programs,” AIAA 57th Int. Astronaut. Congr. IAC 2006, vol. 5, pp. 3219–3233, 2006.

[43] J. Straub, R. Fevig, J. Casler, and O. Yadav, ”Risk analysis & management in student-centered spacecraft development projects,” in Proceedings of the 2013 Reliability and Maintainability Symposium 2013 Jan.

[44] A. Alanazi and J. Straub, “Statistical Analysis of CubeSat Mission Failure,” in AIAA/USU SmallSat Conference, Logan, UT, USA, 4–9 August 2018.

[45] J. F. Castet and J. H. Saleh, “Beyond reliability, multi-state failure analysis of satellite subsystems: A statistical approach,” Reliab. Eng. Syst. Saf., vol. 95, no. 4, pp. 311–322, 2010, doi: 10.1016/j.ress.2009.11.001.

[46] J. F. Castet and J. H. Saleh, “Satellite and satellite subsystems reliability: Statistical data analysis and modeling,” Reliab. Eng. Syst. Saf., vol. 94, no. 11, pp. 1718–1728, 2009, doi: 10.1016/j.ress.2009.05.004.

[47] J. F. Castet and J. H. Saleh, “Single versus mixture Weibull distributions for nonparametric satellite reliability,” Reliab. Eng. Syst. Saf., vol. 95, no. 3, pp. 295–300, 2010, doi: 10.1016/j.ress.2009.10.001.

[48] J. Guo, L. Monas, and E. Gill, “Statistical analysis and modelling of small satellite reliability,” Acta Astronaut., vol. 98, no. 1, pp. 97–110, 2014, doi: 10.1016/j.actaastro.2014.01.018.

[49] J. Straub, J. Berk, A. Nervold, C. Korvald, and D. Torgerson, “Openorbiter: Analysis of a student-run space program,” Proc. Int. Astronaut. Congr. IAC, vol. 13, pp. 9895–9904, 2013.

[50] J. Straub and D. Whalen, “An Assessment of Educational Benefits from the OpenOrbiter Space Program,” Educ. Sci., vol. 3, no. 3, pp. 259–278, 2013, doi: 10.3390/educsci3030259.

[51] J. Straub and D. Whalen, “OpenOrbiter: A low-cost, educational prototype CubeSat mission architecture,” Machines, vol. 1, no. 1, pp. 1–32, 2013, doi: 10.3390/machines1010001.

[52] J. Straub and D. Whalen, “Evaluation of the Educational Impact of Participation Time in a Small Spacecraft Development Program,” Educ. Sci., vol. 4, no. 1, pp. 141–154, 2014, doi: 10.3390/educsci4010141.

[53] J. Praks, A. Kestila, T. Tikka, H. Leppinen, O. Khurshid, and M. Hallikainen, “AALTO-1 earth observation cubesat mission - Educational outcomes,” Int. Geosci. Remote Sens. Symp., vol. 2015-Novem, pp. 1340–1343, 2015, doi: 10.1109/IGARSS.2015.7326023.

[54] ESA, “Fly Your Satellite.” http://www.esa.int/Education/CubeSats_-_Fly_Your_Satellite (accessed Oct. 15, 2019).

[55] H. Carreno-Luengo, A. Camps, P. Via, J. F. Munoz, A. Cortiella, D. Vidal, J. Jane, N. Catarino, M. Hagenfeldt, P. Palomo, and S. Cornara, “3CAT-2: An experimental nanosatellite for GNSS-R Earth observation: Mission concept and analysis,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 9, no. 10, pp. 4540–4551, doi: 10.1109/JSTARS.2016.2574717.

[56] J. Castellví, A. Camps, J. Corbera, and R. Alamús, “3Cat-3/MOTS nanosatellite mission for optical multispectral and GNSS-R earth observation: Concept and analysis,” Sensors (Switzerland), vol. 18, no. 1, 2018, doi: 10.3390/s18010140.

[57] NASA, “ElaNa.” https://www.nasa.gov/content/about-elana (accessed Jan. 15, 2020).

[58] L. Alminde, M. Bisgaard, D. Vinther, T. Viscor, and K. Z. Østergard, “The AAU-Cubesat Student Satellite Project: Architectural Overview and Lessons Learned,” IFAC Proc. Vol., vol. 37, no. 6, pp. 949–954, 2004, doi: 10.1016/s1474-6670(17)32301-7.

[59] J. Guo, J. Bouwmeester, and E. Gill, “From Single to Formation Flying CubeSats: An Update of the Delfi Programme,” Annu. AIAA/USU Conf. Small Satell., pp. SSC13-WK-5, 2013, [Online]. Available: http://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=2903&context=smallsat.

[60] C. J. Fong, A. Lin, M. S. Chang, Y. L. Kuo, A. Tai, B. K. Huang, & C. H. Lee, “NSPO SMART PHONE AND PERSONAL SATELLITE CUBESAT PLATFORMS: YAMSAT-2 & 3,” no. June 2003, pp. 1–9, 2016.

[61] L. Johnson, M. Whorton, A. Heaton, R. Pinson, G. Laue, and C. Adams, “NanoSail-D: A solar sail demonstration mission,” Acta Astronaut., vol. 68, no. 5–6, pp. 571–575, 2011, doi: 10.1016/j.actaastro.2010.02.008.

[62] M. Noca, F. Jordan, N. Steiner, T. Choueiri, F. George, G. Roethlisberger, N. Scheidegger, H. Peter-Contesse, M. Borgeaud, R. Krpoun, and H. Shea, “Lessons learned from the first Swiss pico-satellite: SwissCube,” in the 23rd Annual AIAA/USU SmallSat Conference, Logan, UT, 13 August 2009, paper SSC09-XII-9.

[63] ECSS, “Space product assurance: Quality assurance (ECSS-Q-ST-20C Rev.1),” software engineering handbook, ESA Requirements Standards Division, pp. 1–67, 2013.

[64] ESA, ESTEC, “Tailored ECSS engineering standards for in-orbit demonstration CubeSat projects,”ESA, ESTEC: Noordwijk, The Netherlands, 2016.

[65] ESA, ESTEC, “Distribution External Product and Quality Assurance Requirements for In-Orbit Demonstration CubeSat Projects,” Approved/Applicable Document Type RQ, Issue 1. Revision 1. Date of Issue 26/03/2013.

[66] N. Weitz, “Analysis of Verification and Validation Techniques for Educational CubeSat Programs,” M.S Thesis, College of Computer Science, California Polytechnic State University, San Luis Obispo, CA, USA, 2018. Accessed on January 20, 2019.

[67] P. Fortescue, G. Swinerd, and J. Stark, “Spacecraft Systems Engineering,” 4th ed., John Wiley and Sons Inc., 2011, pp. 540–542.

[68] A. Alanazi, A. B. Jones, and J. Straub, “Requirements Modeling Language and Automated Testing for CubeSats,” in AUTOTESTCON (Proceedings), 2019, vol. 2019-January, doi: 10.1109/AUTOTESTCON43700.2019.8961058.

[69] J. Beatty and A. Chen ”Visual Models for Software Requirements” Redmond, WA: Microsoft Press. ISBN 978-0-7356-6772-3. 2012

[70] H. Reza, R. Sehgal, J. Straub, and N. Alexander, “Toward model-based requirement engineering tool support,” IEEE Aerosp. Conf. Proc., no. October, 2017, doi: 10.1109/AERO.2017.7943647.

[71] S. Friedenthal, A. Moore, R. Steiner, S. Friedenthal, A. Moore, and R. Steiner, “Chapter 3 – Getting Started with SysML,” in A Practical Guide to SysML, 2014.

[72] C. Nieto-Peroy and M. R. Emami, “CubeSat mission: From design to operation,” Appl. Sci., vol. 9, no. 15, pp. 1–24, 2019, doi: 10.3390/app9153110.

[73] I. Karac and B. Turhan, “What Do We (Really) Know about Test-Driven Development?,” IEEE Softw., vol. 35, no. 4, pp. 81–85, Jul. 2018, doi: 10.1109/MS.2018.2801554.

[74] H. Reza, D. Jurgens, J. White, J. Anderson, and J. Peterson, “An architectural design selection tool based on design tactics, scenarios and nonfunctional requirements,” 2005, doi: 10.1109/EIT.2005.1627052.

[75] H. Reza and E. Grant, “Quality-Oriented Software Architecture,” 2005, doi: 10.1109/ITCC.2005.237.

[76] “Visure.” https://visuresolutions.com/systems-engineering (accessed Dec. 05, 2019).

[77] “InteGREAT.” http://www.modernrequirements.com/integreat/ (accessed Dec. 05, 2019).

[78] J. M. Carrillo De Gea, J. Nicolás, J. L. Fernández Alemán, A. Toval, C. Ebert, and A. Vizcaíno, “Requirements engineering tools: Capabilities, survey and assessment,” Inf. Softw. Technol., vol. 54, no. 10, pp. 1142–1157, 2012, doi: 10.1016/j.infsof.2012.04.005.

[79] H. Reza, C. Korvald, J. Straub, J. Hubber, N. Alexander, and A. Chawla, “Toward requirements engineering of cyber-physical systems: Modeling CubeSat,” in IEEE Aerospace Conference Proceedings, 2016, vol. 2016-June, doi: 10.1109/AERO.2016.7500897.

[80] S.C. Spangelo, D. Kaslow, C. Delp, B. Cole, L. Anderson, E. Fosse, B.S. Gilbert, L. Hartman, T. Kahn, and J. Cutler, “Applying model-based systems engineering (MBSE) to a standard CubeSat,” in 2012 IEEE aerospace conference, Big Sky, MT, USA, Mar 3, pp. 1-20. DOI: 10.1109/AERO.2012.6187339.

[81] A. Hartman and K. Nagin, “The AGEDIS Tools for Model-Based Testing,” ACM SIGSOFT Software Engineering Notes, vol. 29, no.4, pp. 129-132, July 2004.

[82] A. Hartman and K. Nagin, “The AGEDIS tools for model-based testing,” Lect. Notes Computer Science, vol. 3297, pp. 277–280, 2005, doi: 10.1007/978-3-540-31797-533.

[83] P. H. Feiler, D. P. Gluch, and J. J. Hudak, “The Architecture Analysis & Design Language (AADL): An Introduction,” no. February, p. CMU/SEI-2006-TN-011, 2006, [Online]. Available: http://www.sei.cmu.edu/library/abstracts/reports/06tn011.cfm.

[84] B. Selic, “Using UML for modeling complex real-time systems,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 1474, pp. 250–260, 1998, doi: 10.1007/BFb0057795.

[85] D. De Niz, “Diagrams and Languages for Model-Based Software Engineering of Embedded Systems: UML and AADL,” pp. 1–9, 2007.

[86] J. Jürjens, “Secure Systems Development with UML,” Springer Science Business Media, pp.55-68, Dec 6, 2005.

[87] J. Jürjens and P. Shabalin, ”Tools for secure systems development with UML: Security analysis with ATPs,” in International Conference on Fundamental Approaches to Software Engineering 2005 Apr 4 (pp. 305-309). Springer, Berlin, Heidelberg.

[88] M. Langer, F. Schummer, N. Appel, T. Gruebler, K. Janzer, J. Kiesbye, L. Krempel, A. Lill, D. Messmann, S. Rueckerl, and M. Weisgerber, “MOVE-II the Munich Orbital Verification Experiment II,” in Advances in the Astronautical Sciences, 2018, vol. 163, pp. 441–459.

[89] M. Martínez, G. Diego, R. Diego, B. Johan, J. Bagur, R. Paz, E. Miranda, and F. Solórzano, “Guatemala’s Remote Sensing CubeSat - Tools and Approaches to Increase the Probability of Mission Success,” in AIAA/USU SmallSat Conference, Logan, UT, July 2018.

[90] D. Kaslow, G. Soremekun, H. Kim, and S. Spangelo, “Integrated model-based systems engineering (MBSE) applied to the Simulation of a CubeSat mission,” in IEEE Aerospace Conference, March 2014, pp. 1-14.

[91] D. Kaslow, B. J. Ayres, P. T. Cahill, C. Croney, L. Hart, and A. G. Levi, “Developing a cubesat model-based system engineering (MBSE) reference model – Interim status #4,” 2018 AIAA Sp. Astronaut. Forum Expo., 2018, doi: 10.2514/6.2018-5328.

[92] D. Kaslow, B. Ayres, P. T. Cahill, L. Hart, and R. Yntema, “Developing a CubeSat Model-Based System Engineering (MBSE) reference model - Interim status #3,” 2017, doi: 10.1109/AERO.2017.7943691.

[93] M. Tolmasoff, R. Delos, and C. Venturini, ”Improving mission success of CubeSats,” InProceedings of the US Space Program Mission Assurance Improvement Workshop, The Boeing Company, El Segundo, CA 2017.

[94] B. Shiotani, N. G. Fitz-Coy, and S. Asundi, “An end-to-end design and development life-cycle for cubesat class satellites,” AIAA Sp. 2014 Conf. Expo., no. August 2014, doi: 10.2514/6.2014-4194.

[95] US Department of Defense Systems Management College, “US Department of Defense Systems Management College,” 22060-5565, no. January, p. 222, 2001, [Online]. Available: http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-885j-aircraft-systems-engineering-fall-2005/readings/sefguide_01_01.pdf.

[96] J. J. Vaske, J. Beaman, and C. C. Sponarski, “Rethinking Internal Consistency in Cronbach’s Alpha,” Leis. Sci., vol. 39, no. 2, pp. 163–173, Mar. 2017, doi: 10.1080/01490400.2015.1127189.

[97] L. J. Cronbach, E. Assistance, and R. J. Shavelson, “My Current Thoughts on Coefficient Alpha and Successor Procedures CSE Report 643,” 2004.

[98] D. George and P. Mallery, “SPSS for Windows Step by Step A Simple Guide and Reference,” Fourth Edition (11.0 update) Answers to Selected Exercises.

[99] A.B. An, “Performing logistic regression on survey data with the new SURVEYLOGISTIC procedure, ” Available: http://www2. sas. com/proceedings/sugi27/p258-27. pdf.” Accessed April 3, 2019.

[100] D. Scheithauer, and K. Forsberg, “4.5. 3 V-Model Views,” in INCOSE International Symposium, Vol. 23, No. 1, pp. 502-516, June 2013.

[101] J. Hudak, and P. Feiler, “Developing AADL models for control systems: A practitioner’s guide,” No. CMU/SEI-2007-TR-014. CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST, 2007.

APPENDIX. SURVEY QUESTIONS

Q1 How do you define a successful CubeSat mission?

———————————————

Q2 What are the most challenging activities in a CubeSat project lifecycle?

———————————————

Q3 What is/was your CubeSat mission?

Scientific
Educational
Commercial
Other ————————

Q4 What is/was the team size of your CubeSat project?

3-6
7-10
11-15
16 or more

Q5 What is your role in the CubeSat project?

Principle Investigator (PI)
Systems Analyst
Software Engineer
Other ————————

Q6 How many CubeSat projects have you participated in?

1-3
4-7
8 or more

Q7 The CubeSat project you are associated with is/was:

Underdevelopment
Incomplete
Failed
Succeed

Q8 The most associated factor of a CubeSat project success is:

HW/SW sufficient testing
COTS HW/SW functionality
Requirements analysis & documentation
Other ————————

Q9 Do most of system designs fail due to tools or models? and Why?

———————————————

Q10 Has your system design reduced testing efforts and/or improved performance?

———————————————

Q11 The most associated factor of a CubeSat project failure is:

HW/SW insufficient testing
COTS HW/SW failure
Lack of requirements analysis and documentation
Other ————————

Q12 The CubeSat you are associated with is:

1U
2U
3U
Other ——————————

Q13 What is the approximate time needed to develop a typical CubeSat?

2 years or less
3-4 years
5-6 years
7 years or more

Q14 The most prominent aspect of software quality that is associated with a CubeSat mission success is:

Reliability
Portability
Modifiability
Usability
Simplicity

Q15 The least prominent aspect of software quality that is associated with a CubeSat mission success is:

Reliability
Portability
Modifiability
Usability
Simplicity

Q16 Which of the following methodologies is most suitable for your CubeSat project? And Why?

Waterfall ——————————
V-Model ——————————
SCRUM ——————————
Other ——————————

Q17 Which technique is best used for requirements elicitation?

Interview
Focus Groups
Prototyping
Other ——————————

Q18 If you are given a second chance to redesign your CubeSat, what would you do instead? and why?

——————————————————————————

Q19 CubeSat mission fail due to failure of tools rather than models.

Strongly disagree
Disagree
Neither agree nor disagree
Agree
Strongly agree

Q20 CubeSat mission fail due to failure of models rather than tools.

Strongly disagree
Disagree
Neither agree nor disagree
Agree
Strongly agree

Q21 Project scope and goals were clearly identified at the early stage of development.

Strongly disagree
Disagree
Neither agree nor disagree
Agree
Strongly agree

Q22 A methodology was carefully selected for our CubeSat project.

Strongly disagree
Disagree
Neither agree nor disagree
Agree
Strongly agree

Q23 Stakeholders can add/delete requirements in the CubeSat testing phase.

Strongly disagree
Disagree
Neither agree nor disagree
Agree
Strongly agree

Q24 Stakeholders participated heavily in the CubeSat design phase.

Strongly disagree
Disagree
Neither agree nor disagree
Agree
Strongly agree

Q25 A project team size has a positive effect on the CubeSat mission.

Strongly disagree
Disagree
Neither agree nor disagree
Agree
Strongly agree

Q26 Requirements were complex and hard to follow.

Strongly disagree
Disagree
Neither agree nor disagree
Agree
Strongly agree

Q27 Requirements were elicited and well documented.

Strongly disagree
Disagree
Neither agree nor disagree

Agree
Strongly agree

Q28 Requirements specification contains several non-testable functional requirements.

Strongly disagree
Disagree
Neither agree nor disagree
Agree
Strongly agree

Q29 The application of methodology has a positive effect on a CubeSat mission.

Strongly disagree
Disagree
Neither agree nor disagree
Agree
Strongly agree

Q30 Applied methodology enhanced the communication and control subsystems.

Strongly disagree
Disagree
Neither agree nor disagree
Agree
Strongly agree

Q31 A CubeSat project had attended to its original scope.

Strongly disagree
Disagree
Neither agree nor disagree
Agree
Strongly agree

Q32 A CubeSat project schedule was well managed.

Strongly disagree
Disagree
Neither agree nor disagree
Agree
Strongly agree