# Ariane 5 Launcher Accident

Swetha Konkyana

March 20, 2023

## 1 Introduction

### 1.1 purpose

- This case study describes the accident that occurred on the initial launch of the Ariane 5 rocket, a launcher developed by the European Space Agency.
- The rocket exploded shortly after take-off and the subsequent enquiry showed that this was due to a fault in the software in the inertial navigation system.
- in June 1996, the then new Ariane 5 rocket was launched on its maiden flight. It carried a payload of scientific satellites.
- Ariane 5 was commercially very significant for the European Space Agency as it could carry a much heavier payload than the Ariane 4 series of launchers.
- Thirty seven seconds into the flight, software in the inertial navigation system, whose software was reused from Ariane 4, shut down causing incorrect signals to be sent to the engines.

### 1.2 Scope

- This case study illustrates issues with requirements specification, multi-organisational working, critical systems validation and some of the problems of software reuse.
- The example illustrates that good software engineering practice (reuse, don't introduce changes unless necessary) can have problems and highlights the need for diversity as well as redundancy.
- It also shows the organisational complexity of systems development and how organisational issues can lead to systems failure.
- I have used it in conjunction with lectures on critical systems validation.

### 1.3 Problem Definition

- the very first Ariane 5 rocket ignited its engines and began speeding away from the coast of French Guiana.
- 37 seconds later, the rocket flipped 90 degrees in the wrong direction, and less than two seconds later, aerodynamic forces ripped the boosters apart from the main stage at a height of 4km.
- This caused the self-destruct mechanism to trigger, and the spacecraft was consumed in a gigantic fireball of liquid hydrogen.

### 1.4 System Overview

For the readers the document is intended for, people such as developers, project managers, marketing staff, users, testers, and documentation writers, the "client" and the professor.

## 1.5 References

- https://iansommerville.com/software-engineering-book/case-studies/ariane5/
- http://www.rvs.uni-bielefeld.de/publications/Reports/ariane.html
- http://sunnyday.mit.edu/accidents/Ariane5accidentreport.htm
- https://www.dropbox.com/s/tr02fmo4xzytzhv/Bashar-Ariane5.pdf?dl=0
- www.google.com

# 2 Overall Description

## 2.1 Product Perspective

The Product is Software Designed to Launch a Rocket . Constitutes Internal Calculations that help in maintaining the rocket in the right parameter of flight.

Scientific, Development and Maintenance Team Interacts with the software.

## 2.2 Product Functions

Product provides Calculations like Ballastic Flight, Terminal Velocity, Rocket Altitude, Mach and Speed of Sound.

Each has it's own importance in launching a rocket.

## 2.3 User Characteristics

1. Home page
2. Ballistic Flight Calculator
3. Terminal Velocity Calculator
4. Rocket Altitude Calculator
5. Mach and Speed ofsound Calculator

## 2.4 Constraints

1. As the product is inherited from the Case Study Of Ariane – 5 Launch Accident, the rocket has blasted in 37 secs into the flight after launch.
2. Reason being no proper code reuse.
3. The context didn't provided enough information about how we can develop a complete end to end rocket launching software.
4. Also many companies keep their source code confidential, making it much more difficult to get ideas related to it.
5. Understanding how various scientific terms are involved and how they can be merged into one so that they can led to a meaningful outcomes is a big task.

## 2.5 Assumption

- Any of the modern Operating Systems are Supported.
- Windows, Linux, Mac are basic OS that are fully supported.
- Browser should be of the lastest generation.
- Browsers like Internet Explorer are not supported.

## 2.6 Dependencies

- An active internet connection is highly recommended.
- No further packages or software plugins are required.

# 3 Specific Requirements

## 3.1 External Interfaces

### 3.1.1 User Interfaces

1. HOME PAGE

   For providing navigation buttons to different features in the software

   DESCRIPTION :

   This home page is designed for showing the features of the software designed The software designed works on calculating various parameters for launching the Software

   FEATURES:

   1.Ballistic flight calculator 2.terminal velocity calculator 3.rocket altitude calculator 4. mach speed sound calculator

   USER INTERACTION AND RESPONSE:

   Clicking upon any navigation button the user is directed for navigation to the respective page in the software

2. BALLISTIC FLIGHT CALCULATOR:

   DESCRPTION:

   This program calculates the maximum height of a launched ballistic shell, or a shell with drag, and the time from launch when the maximum height is reached.

   FEATURES

   · Initial velocity · Mass · Cross section area · Drag coefficient · Altitude etc.

   USER INTERACTION AND RESPONSE:

   User will input numerical values and output is shown Further user can use RAC and home navigation buttons.

3. TERMINAL VELOCITY CALCULATOR:

   DESCRIPTION:

   This program calculates the terminal velocity of a falling object. Circular Orbit Calculator This program calculates the altitude and velocity of an object in a circular orbit about the Earth, Moon or Mars.

   FEATURES:

   · Mass · Cross section area · Drag coefficient · Altitude etc.

   USER INTERACTION AND RESPONSE:

   User will input numerical values and output is shown Further user can use RAC and home navigation buttons.

### 3.1.2 Hardware Interfaces

- All modern day computers with 32bit or 64bit Architecture supports the software.
- Network Interface Card or a RJ – 45 port are mandatory.
- CPU processor above i3 8th gen, i5 8th gen are recommend.
- Ryzen series of ¿= 4000 are recommend.

- Standard Keyboard and respective drivers are required.
- Standard Mouse Pointer and respective drivers are required.
- All modern Android Versions are Supported.
- A RAM of atleat 6GB is recommend.

#### 3.1.3 Software Interfaces

The list of interfaces implemented in the software are:

1. Home Page.
2. Ballastic Flight Calculator.
3. Terminal Velocity Calculator.
4. Rocket Altitude Calciator.
5. Mach and Speed of Sound Calculator

### 3.2 Functions

- Taking Precise Inputs form the user.
- Showing processed Calculations and Output to the User.
- Providing navigation to other interfaces wherever required.

#### 3.2.1 F1: The system shall take precise Inputs form the user.

#### 3.2.2 F2: The system shows processed Calculations and Output to the User.

#### 3.2.3 F3: The system shows provide navigation to other interfaces wherever required.

### 3.3 Performance Requirements

- If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices.
- Specify the timing relationships for real time systems.
- Make such requirements as specific as possible.
- You may need to state performance requirements for individual functional requirements or features.
- TO DO: Provide performance requirements based on the information you collected from the client/professor.
- For example, you can say "P1. The secondary heater will be engaged if the desired temperature is not reached within 10 seconds"

### 3.4 Logical database Requirements

1. Lower composite (EAP and EPC)
2. Upper composite (ESC-A, VEB and supporting structure)
3. Launcher upper part (fairing and Sylda 5)

### 3.5 Design Constraints

- As the product is inherited from the Case Study Of Ariane – 5 Launch Accident, the rocket has blasted in 37 secs into the flight after launch.
- Reason being no proper code reuse.
- The context didn't provided enough information about how we can develop a complete end to end rocket launching software.

### 3.6 Key Features

- Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product.
- Define any safeguards or actions that must be taken, as well as actions that must be prevented.
- Refer to any external policies or regulations that state safety issues that affect the product's design or use.
- Define any safety certifications that must be satisfied.
- Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product.
- Define any user identity authentication requirements.

# 4 Validation Check

## 4.1 Validity check

- Validation failure As the facility that failed was not required for Ariane 5, there.
- Review failure The design and code of all software should be reviewed.
- The functions proposed by stakeholders should be aligned with what the system needs to perform.
- You may find later that there are additional or different functions are required instead.

## 4.2 Consistency check

- Requirements in the document shouldn't conflict or different descriptions of the same function
- Consistency check for Ariane 5 is a process of verifying that the various systems and components of the rocket are working together as expected to ensure a successful launch.
- This includes checking that the software code is correct and has been properly integrated into the overall system, and that all sensors, instruments, and equipment are calibrated and functioning correctly.
- A consistency check is typically performed throughout the development and testing phases of the rocket, as well as during the final countdown before launch.
- It is a critical component of ensuring the safety and success of the mission.

### 4.3 Realism check

- No matter how rigorous we are, software is going to be faulty
- Testing represent a substantial percentage of software development costs and time to market
- Impossible to test under all operating conditions – based on incomplete testing, we must gain confidence that the system has the desired behavior
- Testing large systems is complex – it requires strategy and technology- and is often done inefficiently in practice

### 4.4 Verifiability check

- Requirements should be written so that they can be tested.
- This means you should be able to write a set of tests that demonstrate that the system meets the specified requirements.
- Verifiability for Ariane 5 refers to the process of ensuring that the rocket's design and components are tested and proven to be reliable and safe before launch.
- This is achieved through rigorous testing and validation processes, as well as close collaboration between the rocket's designers, engineers, and manufacturers to ensure that all components meet the necessary specifications and standards.
- In addition to testing the design and components of the rocket, verifiability also involves ensuring that the launch procedures and systems are in place to ensure a successful launch and mission.
- This includes testing the launch site, ground support equipment, and communication systems, as well as conducting simulations and practice runs to identify and mitigate potential issues.
- Overall, verifiability is a critical component of Ariane 5's success, ensuring that the rocket is safe, reliable, and fully prepared for its mission before launch.

### 4.5 Completeness check

- The document should include all the requirements and constraints.
- Completeness check for Ariane 5 would involve verifying that all the required components and systems are present and in working order.
- This includes checking that the rocket's structure is intact, the engines, fuel tanks, electrical systems, avionics, guidance systems, and payload are all present and properly installed. It would also involve checking that all necessary safety measures, such as backup systems and safety mechanisms, are in place and functioning correctly.
- Finally, a completeness check would involve ensuring that all the required documentation and certifications, including launch permits and safety clearances, are complete and up-to-date.

## 5 Validation Techniques

### 5.1 Requirement Reviews

- Arianespace's Ariane 5 is the world reference for heavy-lift launchers, capable of carry payloads weighing more than 10 metric tons to geostationary transfer orbit (GTO) and over 20 metric tons into low-Earth orbit (LEO) – with a high degree of accuracy mission after mission.
- Developed by under management of the European Space Agency (ESA), Ariane 5 is able to loft the heaviest spacecraft either in production or on the drawing boards, and enables Arianespace to match up most telecommunications satellites for highly efficient dual launches – a capability that has been proven by the company in Ariane-series missions since the 1980s.

## 5.2 Prototyping Techniques

- We've discussed prototyping as one of the (non-standalone) software process methodologies which is used as part of a full methodology, and we've also mentioned it can be used in requirements engineering.
- In this approach to validation, an executable model of the system is demonstrated to the customer and end-users to validate and ensure if it meets their needs.
- Prototyping is usually used when the requirements aren't clear.
- So, we make a quick design of the system to validate the requirements. If it fails, we then refine it, and check again, until it meets the customer's needs.
- This definitely will decrease the cost as a result of having clear, understandable, consistent requirements.

## 5.3 Test case Generation

- As we've just mentioned, the requirements need to be testable. If the tests for the requirements are added as part of the validation process, this often reveals requirements problems.
- If the test is difficult or impossible to design, this usually means that the requirements will be difficult to implement and should be reconsidered.