JURUSAN TEKNOLOGI INFORMASI

Software Engineering Course

02. Introduction to Software Engineering

Yoppy Yunhasnawa, S.ST., M.Sc.

Topics

- What is Software Engineering?
- The Importance of Software Engineering
- Software Product

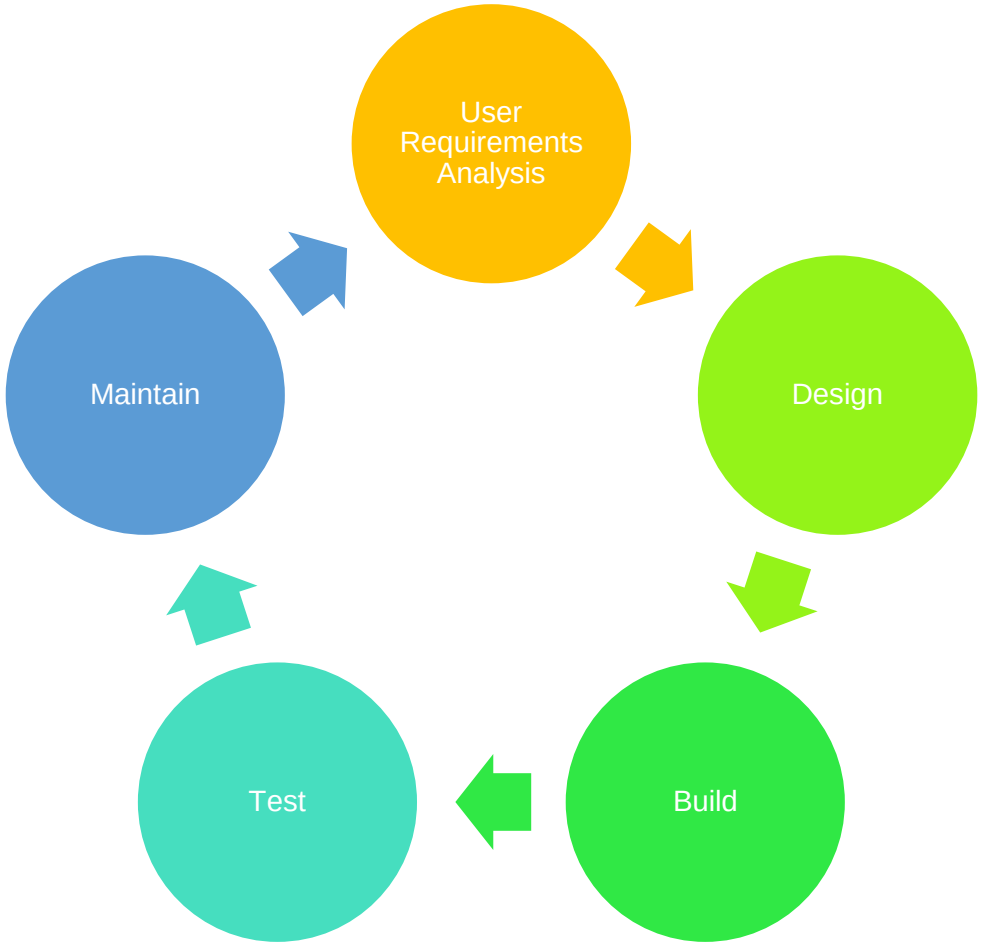
Topic #1: What is Software Engineering

1. What is Software Engineering?

- *"Software engineering is an **engineering discipline** that is concerned with all aspects of software production."* – Ian Sommerville. [1]
- *"Software engineering is the application of a systematic, disciplined, which is **a computable approach** for the development, operation, and maintenance of software."* – IEEE 610.12-1990.
- *"Software engineering is the establishment and used **standard** engineering principles. It helps you to obtain, economically, software which is reliable and works efficiently on the real machines."* - Fritz Bauer.
- *"Software Engineering is the **practical application of scientific knowledge** to the creative design and building of computer programs. It also includes associated documentation needed for developing, operating, and maintaining them."* – Boehm.

To Conclude…

- Software engineering is defined as a process of:
  - Analyzing **user requirements** and then;
  - **Designing**,
  - **Building**, and;
  - **Testing** software application which will satisfy those requirements, as well as;
  - Optionally **maintaining** it.



Why?

- Once upon a time, there was an event famously known as: **Software Crisis**. [2]
- It was in the late 1960s when many software projects failed.
- Many software became **over budget**.
  - Output was an unreliable software which is expensive to maintain.
- Larger software was **difficult and quite expensive to maintain**.

- Lots of software **not able to satisfy the growing requirements** of the customer.
- **Complexities** of software projects increased whenever its hardware capability increased.
- **Demand for new software increased** faster compared with the ability to generate new software.
- All the above issues lead to 'Software Crisis.'

Why?

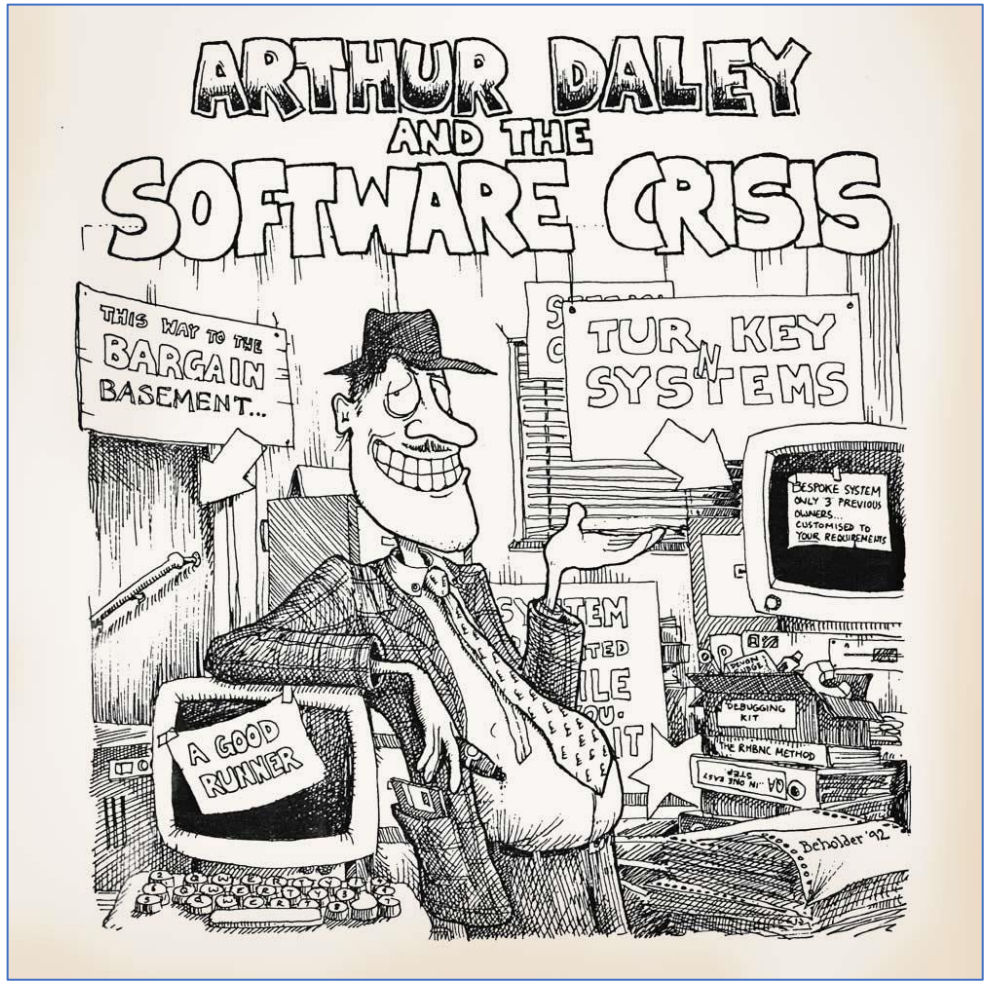


Figure: Software Crisis Depiction. [3]

Solution to The Software Crisis

- The solution was: Transforming unorganized coding effort into a **software engineering discipline**.
- These engineering models helped companies to:
  - Streamline operations
  - Deliver software meeting customer requirements.
- The late 1970s saw the widespread uses of software engineering principles.
- In the 1980s saw the automation of software engineering process and growth of (CASE) Computer-Aided Software Engineering.
- The 1990s have seen an increased emphasis on the 'management' aspects of projects standard of quality and processes just like ISO 9001

Topic #2: The Importance of Software Engineering

2. The Importance of Software Engineering

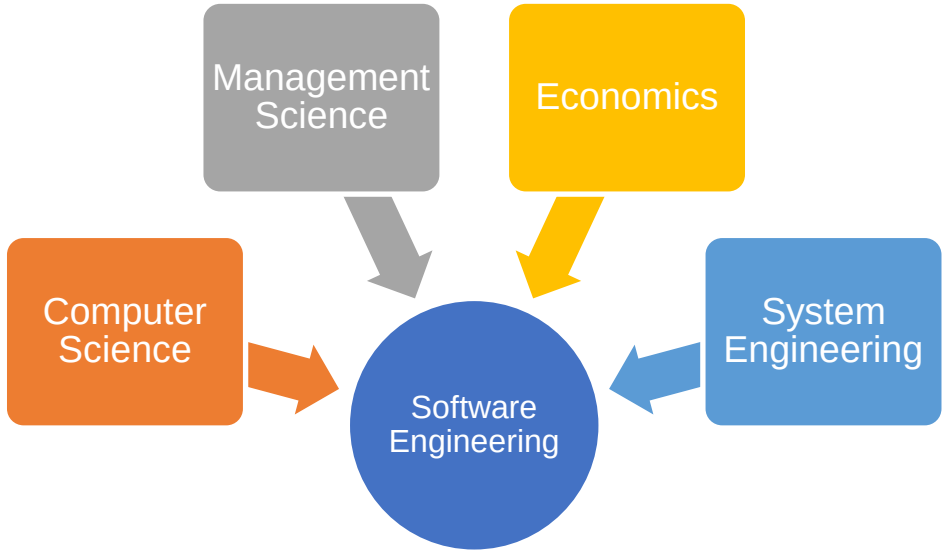
- *Here are important reasons behind the popularity of software engineering: [4]*

2. The Importance of Software Engineering

- **Large software:** In our real life, it is quite *more comfortable to build a wall than a house or building*. In the same manner, as the size of the software becomes large, software engineering helps you to build software.
- **Scalability:** If the software development process were based on scientific and engineering concepts, it is *easier to re-create new software to scale an existing one*.
- **Adaptability**: Whenever the software process was based on scientific and engineering, it is easy to re-create new software with the help of software engineering.
- **Cost:** Hardware industry has shown its skills and huge manufacturing has *lower the cost of the computer and electronic hardware*.
- **Dynamic Nature**: Always *growing and adapting* nature of the software. It depends on the environment in which the user works.
- **Quality Management**: Offers better method of software development to *provide quality software products*.

Relationship of Software Engineering with Other Disciplines

- **Computer Science**: Gives the scientific foundation for the software as electrical engineering mainly depends on physics.
- **Management Science**: Software engineering is labor-intensive work which demands both technical and managerial control. Therefore, it is widely used in management science.
- **Economics**: In this sector, software engineering helps you in resource estimation and cost control. Computing system must be developed, and data should be maintained regularly within a given budget.



- **System Engineering**: Most software is a component of a much larger system. For example, the software in an Industry monitoring system or the flight software on an airplane. Software engineering methods should be applied to the study of this type of systems.

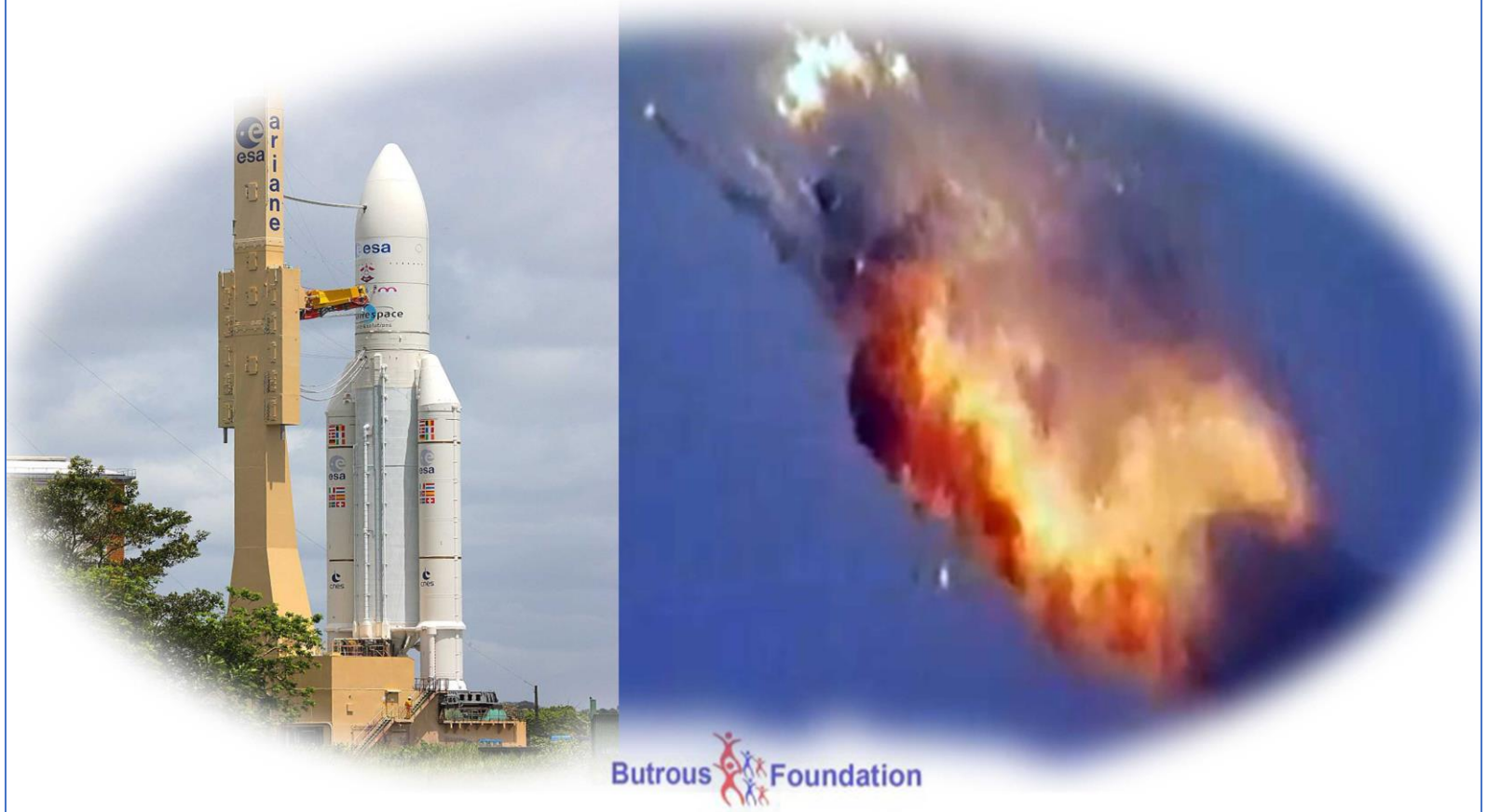
Challenges

- In safety-critical areas, the cost of software failure can be **massive** because lives are at risk. Those are areas such as:
  - Space
  - Aviation
  - Nuclear power plants
  - etc.
- Increased market demands for fast turnaround time.
- Dealing with the increased complexity of software need for new applications.
- The diversity of software systems should be communicating with each other.

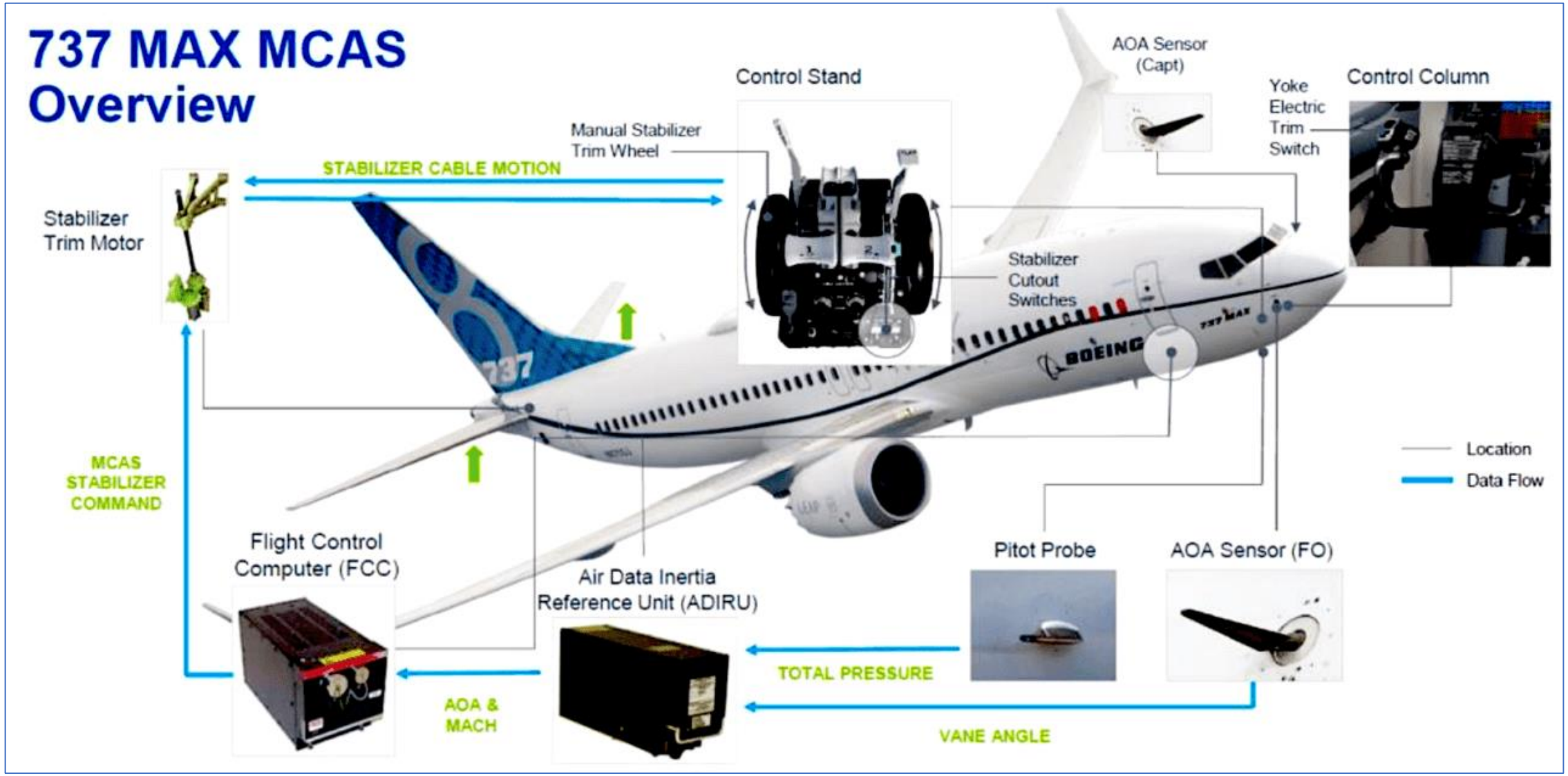
Challenges

4 June 1996

Ariane 5 rocket launched by the European Space Agency exploded just under forty seconds after its lift-off



Challenges


737 MAX MCAS Overview
AOA Sensor (Capt)
Control Stand
Yoke Electric Trim Switch
Control Column
Manual Stabilizer Trim Wheel
STABILIZER CABLE MOTION
Stabilizer Trim Motor
Stabilizer Cutout Switches
MCAS STABILIZER COMMAND
Location
Data Flow
Flight Control Computer (FCC)
Air Data Inertia Reference Unit (ADIRU)
Pitot Probe
AOA Sensor (FO)
TOTAL PRESSURE
AOA & MACH
VANE ANGLE

Topic #3: Software Product

3. Software Product

- The immediate result of the software engineering process is a software. And the quality of the software is judged based on its **characteristics**.

- The characteristics of any software product include features which are displayed by the product when it is installed and put in use.

- They are not the services which are provided by the product. Instead, they have related to:
  - the products dynamic behavior and;
  - the use made of the product.

- Examples of these attributes are:
  - Efficiency,
  - Reliability,
  - Robustness,
  - Maintainability,
  - etc.

- However, the relative importance of these characteristics varies from one software system to another.

Attributes of Software Products

Product Characteristics	Description
Maintainability	The software should evolve to meet the changing demands of the clients.
Dependability	Dependability includes various characteristics. Dependable software should never cause any physical or economic damage at the time of system failure.
Efficiency	The software application should overuse system resources like memory and processor cycle.
Usability	The software application should have specific UI and documentation.

- Optimizing the above attribute is very challenging. For example, offering a better UI can reduce system efficiency.

Characteristics of Good Software

- Any software should be judged by what it offers and what are the methods which help you to use it.

- Every software must satisfy the following attributes:
  - Operational
  - Transitional
  - Maintenance

Characteristics of Good Software: (1) Operational

This characteristic let us know about how well software works in the operations which can be measured on:

- Budget
- Efficiency
- Usability
- Dependability
- Correctness
- Functionality
- Safety
- Security

Characteristics of Good Software: (2) Transitional

This is an essential aspect when the software is moved from one platform to another:

- Interoperability
- Reusability
- Portability
- Adaptability

Characteristics of Good Software: (3) Maintenance

This aspect talks about how well software has the capabilities to adapt itself in the quickly changing environment:

- Flexibility
- Maintainability
- Modularity
- Scalability

Questions?

Thank You

Task

- Please try to find an article or news about a fatal failure on software engineering process.
- Describe about:
  - What happened?
  - Why did it happen?
  - What were the solutions?
- Summarize it into a PPT with at least 3 pages.
- Submit the PPT on the Google Classroom.
- Classroom Code: 6axztdt

References

[1] SOFTWARE ENGINEERING Ninth Edition by Ian Sommerville; Page No. 6.

[2] https://medium.com/@ryancohane/has-the-software-crisis-passed-d45ce975a1e7

[3] https://www.beholder.uk/scrapbook/software-crisis

[4] https://www.guru99.com/what-is-software-engineering.html