FlySense

Standards and Regulations

26th March 2018

Shivang Baveja
Nick Crispie
Joao Fonseca Reis
Harikrishnan Suresh
Sai Nihar Tadichetty

ASTM F3002

Standard for Command and Control System for Unmanned Aircraft

Motivation and Applicability

- **Need a general standard for flying unmanned aerial vehicles**
  - What happens when there is a lost connection?
  - What happens when the aerial vehicle flies somewhere it is not supposed to go?

- **ASTM 30002 attempts to address these questions and enforces a standard for all unmanned aerial vehicles under 25kg ("small" Unmanned Aerial Systems)**

Terminology and Abbreviations

Unmanned Aerial System (UAS):
Flight hardware and control systems for fully functional flying vehicle under 25kg

Ground Control Station (GCS):

Location with equipment for human control of the UAS

Command and control links (C2):

Safety critical RF link between GCS and unmanned aircraft

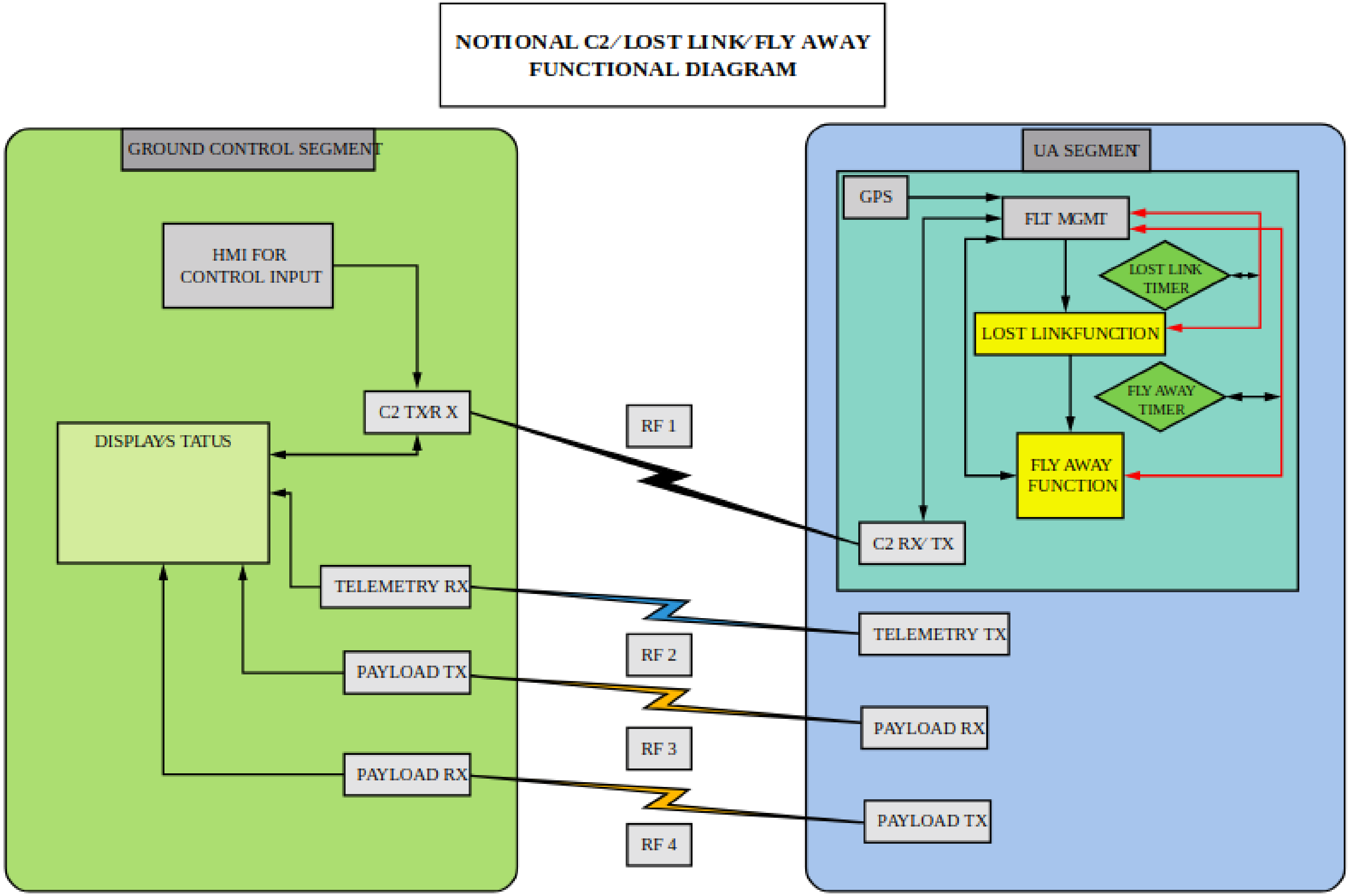
Lost Link:

Condition where the pilot can no longer control the UAS due to loss, interruption, or degradation of signal

Fly Away:

Unintended flight outside of operational area due to failure of control and/or onboard system

Functional Architecture



ASTM F3002: STANDARD FOR COMMAND AND CONTROL SYSTEM FOR UNMANNED AIRCRAFT

General Regulations

- **System must minimize Radio-Frequency interference to ensure robustness of the controller**

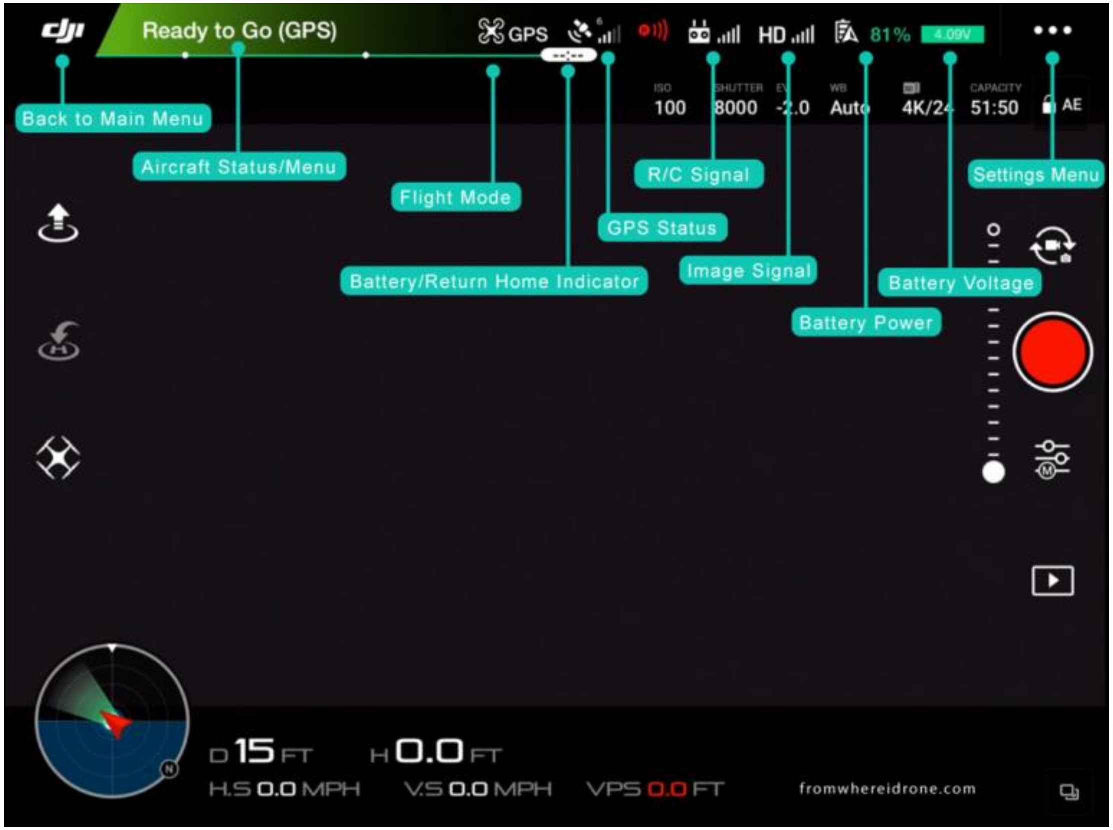
- **Control equipment must be protected from environmental conditions**

- **Control equipment and communication must we rigidly attached to respective hardware and have a robust construction**

THAT SHOULD TAKE CARE OF MY ELECTROMAGNETIC RADIATION PROBLEM!!
ED 4

Ground Control Station Requirements

- **Ground Control Station must give the operator knowledge of the communications status**
- **Ground Control Station must provide telemetry if the aerial vehicle is able to broadcast it**



ASTM F3002: STANDARD FOR COMMAND AND CONTROL SYSTEM FOR UNMANNED AIRCRAFT

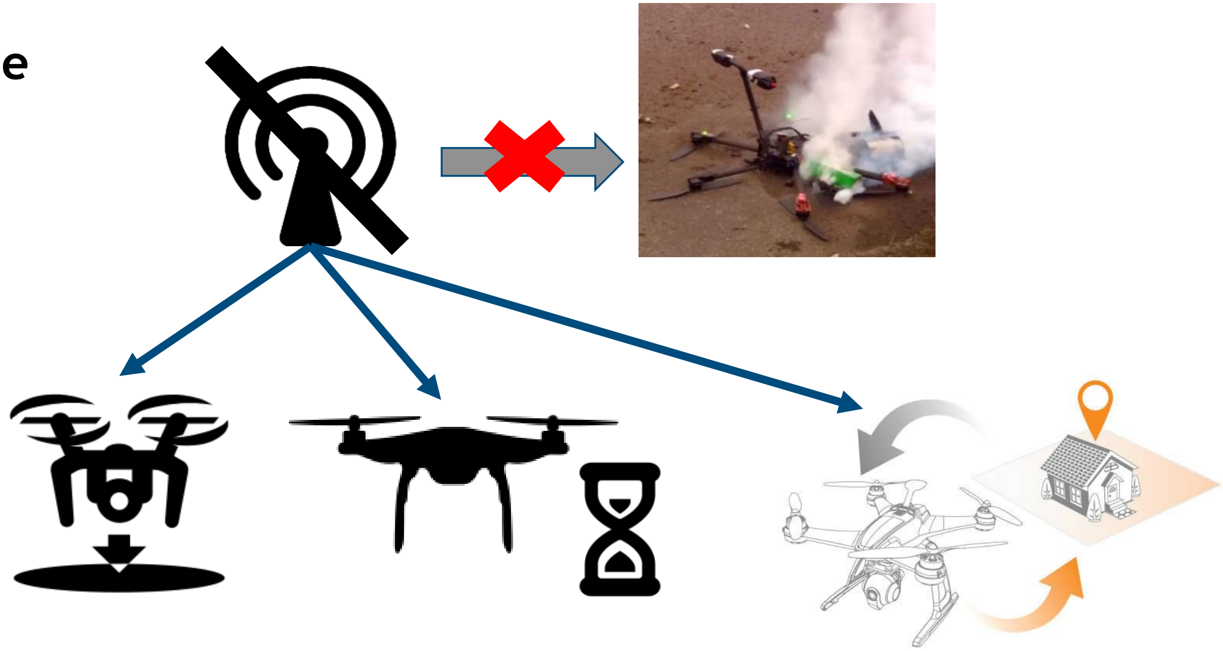
Unmanned Aircraft Requirements

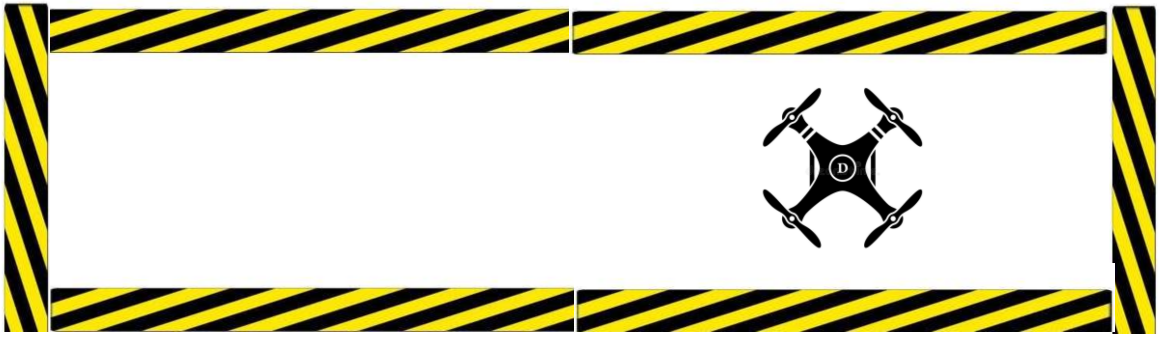
- **Lost link action has to be able to execute even after the loss of communication**

- **Lost Link can’t make the aerial system fail**

- **With loss of communication, aerial system must**
  - Land safely and terminate flight, or
  - Return to home position, or
  - Wait for a time before executing one of the previous options

- **UAS has to be able to operate in a confined area**





Fly Away Requirements

- **Fly away prevention must still work even if communication fails**

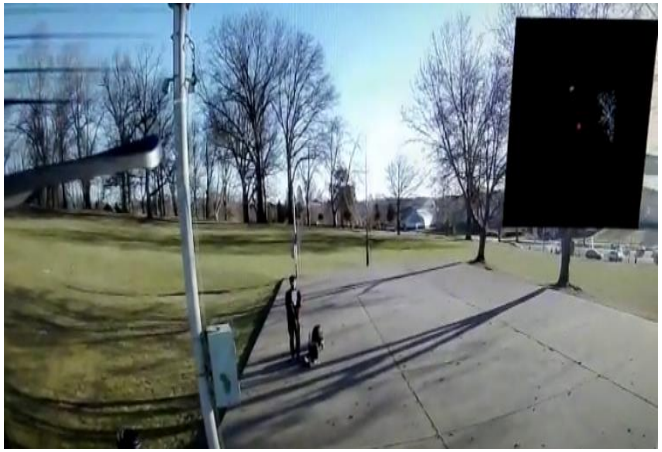
ASTM F3002: STANDARD FOR COMMAND AND CONTROL SYSTEM FOR UNMANNED AIRCRAFT

What it means for FlySense

- **Equipment we use**
  - DJI follows these guidelines as a manufacturer – our system is safety ensured

- **Modifications we make to the software system must comply with the standard**
  - As we make modifications to the system, we can’t override the existing safety and communication architecture

- **How FlySense follows the standard**
  - Take-off only after thorough check with the DJI SDK app
  - One operator is always in-charge of ensuring proper communication
  - Pilot command override for obstacle avoidance implemented using the permitted DJI SDK ‘function mode’

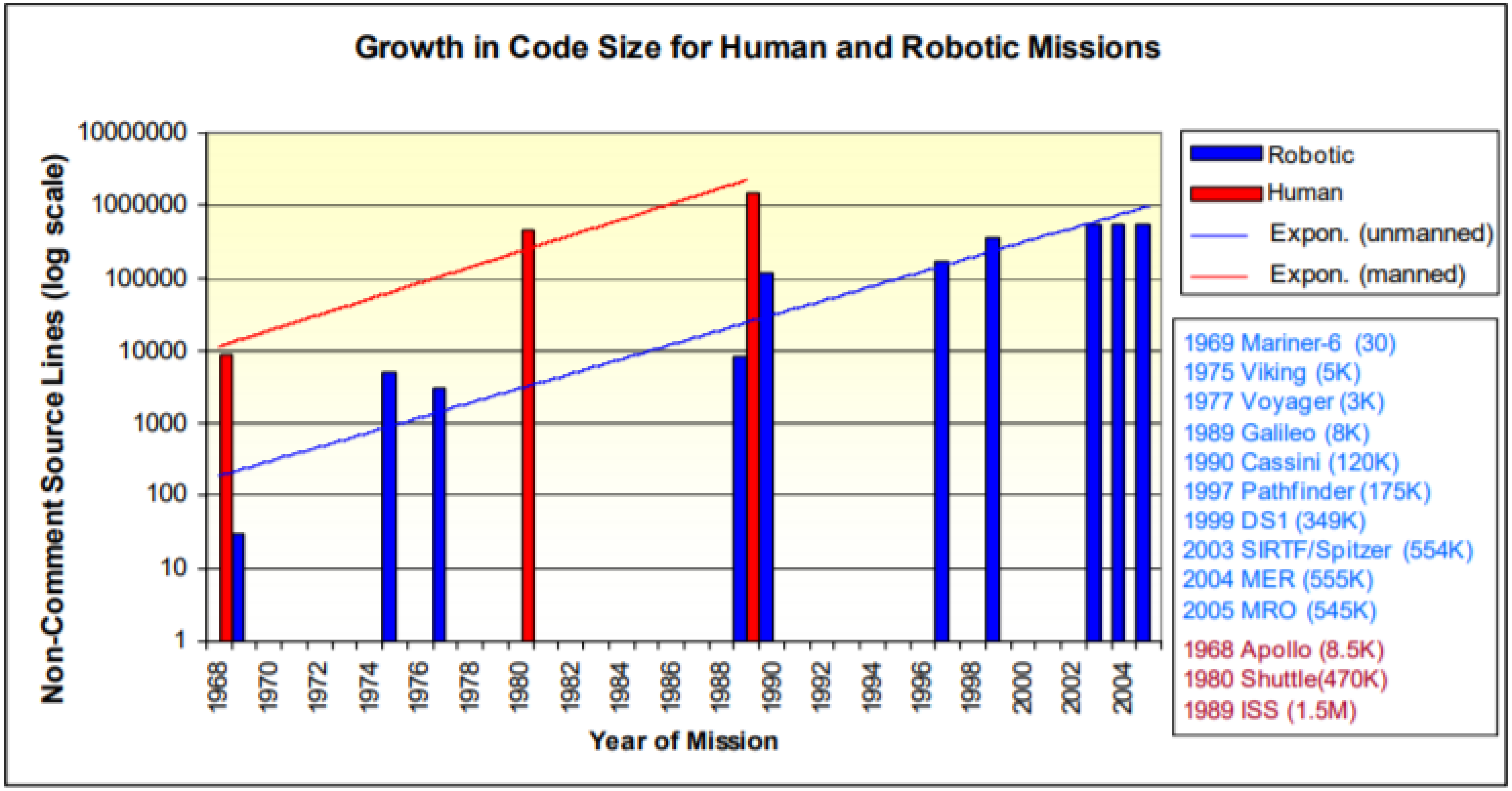


DO-178C

Software Considerations in Airborne Systems and Equipment Certification

DO 178C: WHY WAS IT DEVELOPED

The rapid increase in the use of software in airborne systems resulted in the need for industry-accepted guidance for satisfying airworthiness requirements



History of flight software growth in human and robotic missions

It provides guidelines for the production of software for airborne systems

- **It is the primary document by which the certification authorities such as FAA, EASA and Transport Canada approve all commercial software-based aerospace systems.**

- **A means of showing compliance with the applicable airworthiness regulations for the software aspects of airborne systems and equipment certification.**

Aérospatiale/BAC Concorde is a British-French turbojet

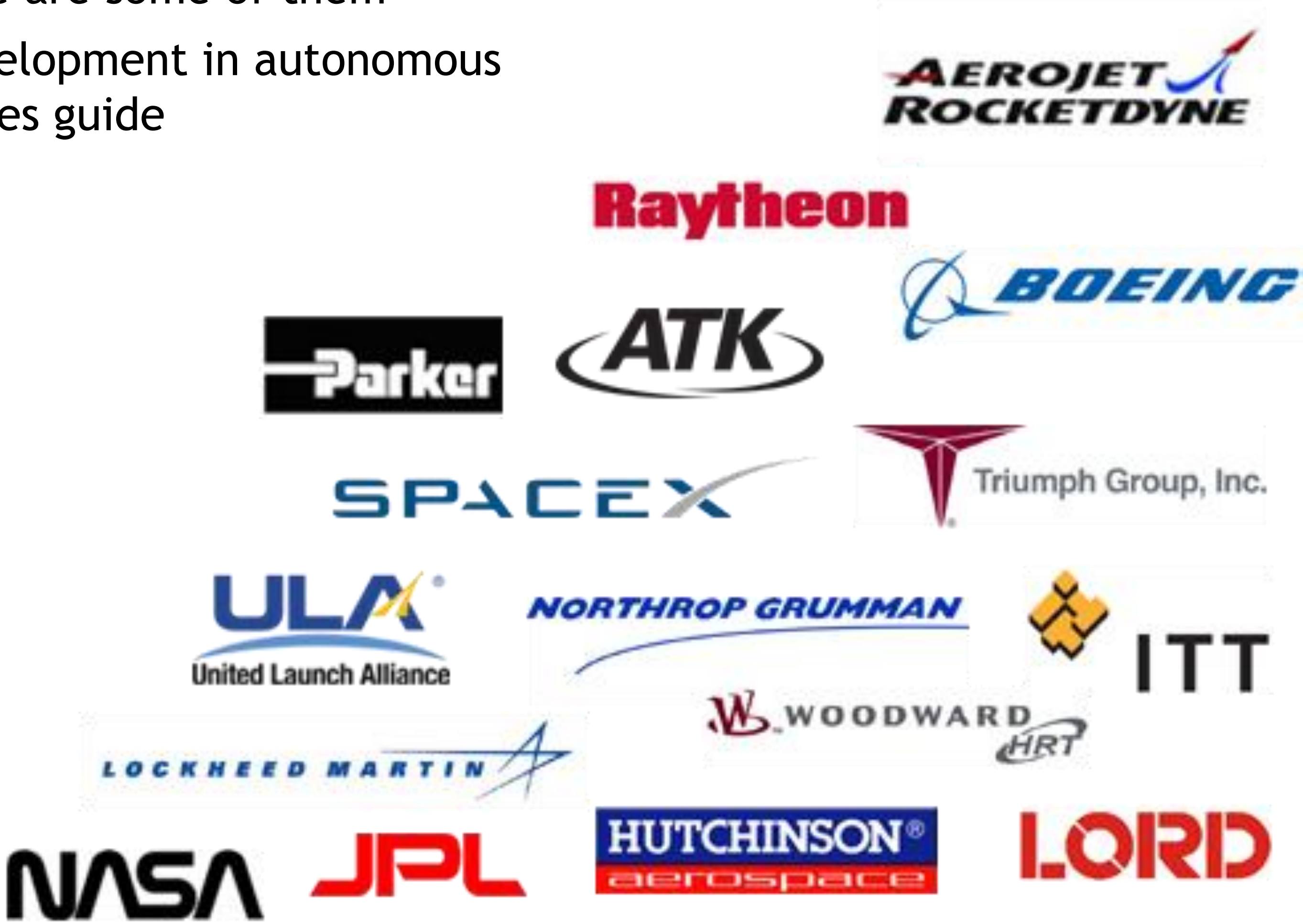
ProjectFolder-AreaFolder-Name-Date-DesktopPublishing-Author

DO 178C:WHO USES THIS STANDARD

Aerospace companies following DO-178 guidelines

- The list is huge, these are some of them
- Also for software development in autonomous cars as a best-practices guide

FlySense

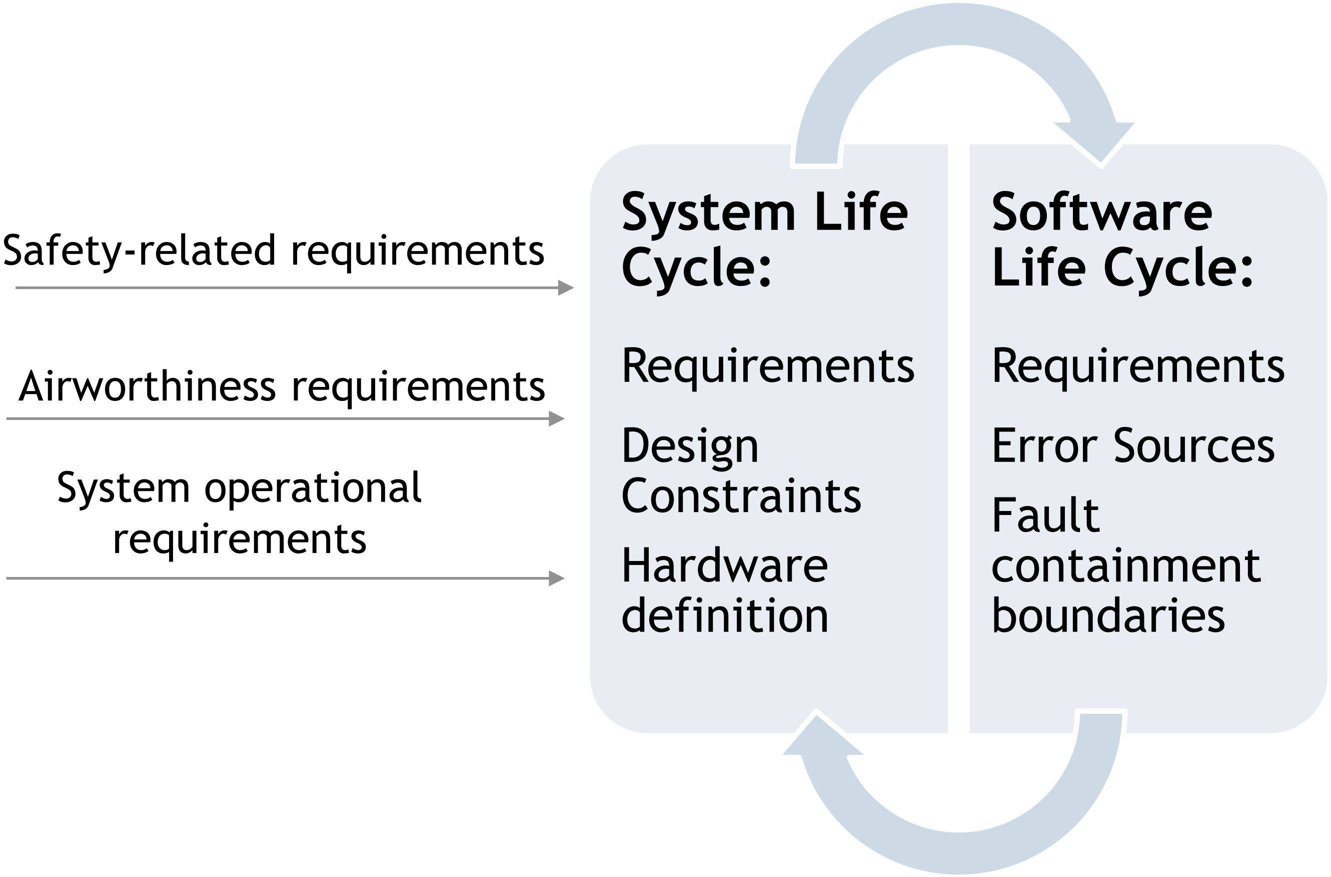


DO 178C: GUIDELINES

What do the guidelines prescribe?

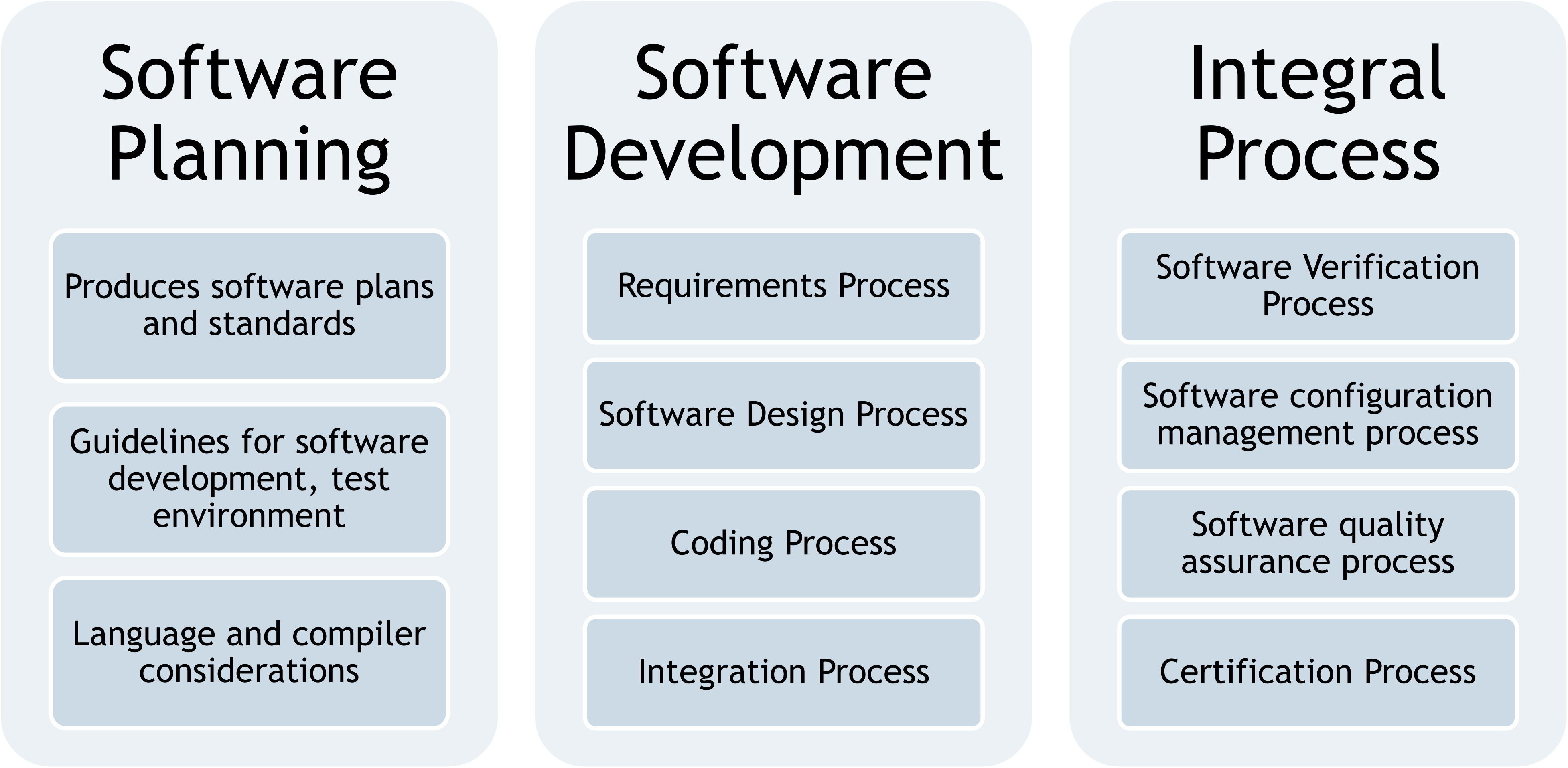
- Objectives for software life cycle processes
- What activities and procedures to follow to achieve those objectives
- What documentation to produce as evidence that the objectives have been satisfied
- Certification issues are discussed only in relation to software life cycle. The operational aspects of the resulting software are not discussed in the document

System <-> Software life cycle


Safety-related requirements
Airworthiness requirements
System operational requirements
System Life Cycle:
Requirements
Design Constraints
Hardware definition
Software Life Cycle:
Requirements
Error Sources
Fault containment boundaries

DO-178C: SOFTWARE LIFE CYCLE PROCESS

Following a predefined process ensures safe and bug free software



Software Planning

- Produces software plans and standards
- Guidelines for software development, test environment
- Language and compiler considerations

Software Development

- Requirements Process
- Software Design Process
- Coding Process
- Integration Process

Integral Process

- Software Verification Process
- Software configuration management process
- Software quality assurance process
- Certification Process

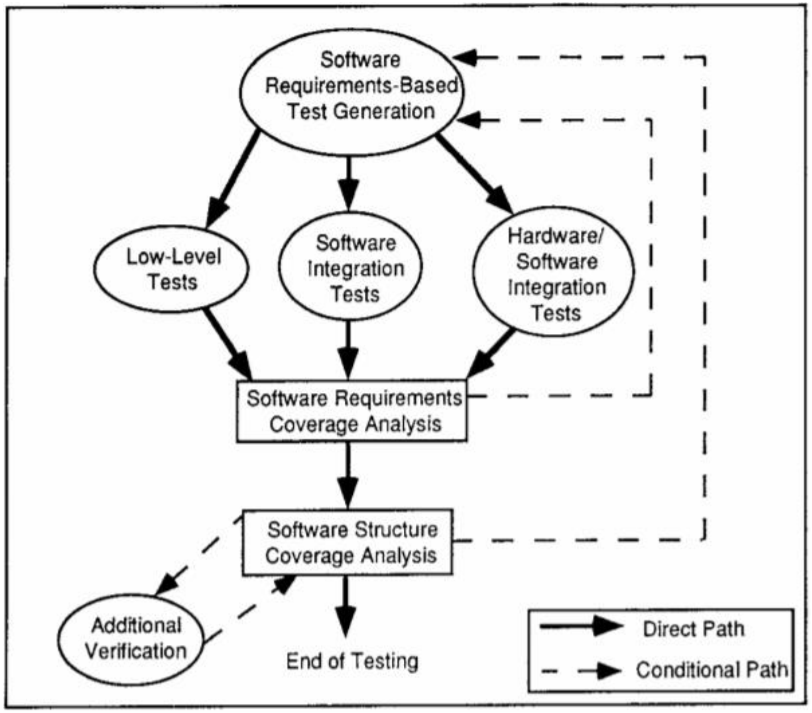
ProjectFolder-AreaFolder-Name-Date-DesktopPublishing-Author

DO-178C: SOFTWARE VERIFICATION

The general objectives of the software verification process are to verify that:

- The system requirements have been developed into software high-level requirements
- The high-level requirements have been developed into software architecture and low-level requirements
- The software architecture and low-level requirements have been developed into Source Code
- The Executable Object Code satisfies the software requirements.
- The means used to satisfy these objectives are technically correct and complete for the software level.

Software testing process


Software Requirements-Based Test Generation
Low-Level Tests
Software Integration Tests
Hardware/ Software Integration Tests
Software Requirements Coverage Analysis
Software Structure Coverage Analysis
Additional Verification
End of Testing
Direct Path
Conditional Path

Goal: Traceability and Correctness

DO-178C: FAILURE CONDITION AND SOFTWARE LEVEL

Higher level implies higher level of effort required to show compliance with certification requirements

Failure Categories	Software Level	Description
Catastrophic	Level A	Prevent continued safe flight and landing
Hazardous/Severe -Major	Level B	Failure conditions which would reduce the capability of the aircraft or potential fatal injuries
Major	Level C	A significant reduction in safety margins or functional capabilities or discomfort to occupants
Minor	Level D	Would not significantly reduce aircraft safety, and which would involve crew actions that are well within their capabilities
No Effect	Level E	Failure conditions which do not affect the operational capability

ProjectFolder-AreaFolder-Name-Date-DesktopPublishing-Author

How does it apply to flysense?

- **Information provided to the pilot has to be accurate, otherwise it can lead to problems**
- **Software has to work in all possible conditions without failure**
- **Pilot override feature is safety critical and should be developed following the guidelines of design, verification and testing**
- **To be DO-178C compliant, we would have to:**
  - Write High-level and low-level software requirements
  - Develop software development plan
  - Ensure traceability of each requirement to the code
  - Establish verification strategies
  - Generate artifacts that prove compliance