Infotech@Aerospace 2012
19 - 21 June 2012, Garden Grove, California

AIAA 2012-2572

An Integrated Safety and Systems Engineering Methodology for Small Unmanned Aircraft Systems

Ewen Denney[*] and Ganesh Pai[†]
SGT Inc., NASA Ames Research Center, Moffett Field, CA 94035, USA

Corey Ippolito[‡]
NASA Ames Research Center, Moffett Field, CA 94035, USA

Ritchie Lee[§]
Carnegie Mellon University, Silicon Valley, Moffett Field, CA 94035, USA

This paper presents an integrated methodology for addressing safety concerns during the systems engineering of small Unmanned Aircraft Systems (sUAS). We describe both the systems and safety engineering activities performed, their interrelations, and how they complement range safety analysis. The broad goal of this work is to support the derivation of an airworthiness statement, and the subsequent application for a Certificate of Authorization (COA) to operate sUAS in the National Airspace System (NAS). We exemplify our methodology by presenting its application to the Swift UAS and its payload data system, both of which are under development at NASA Ames Research Center.

I. Introduction

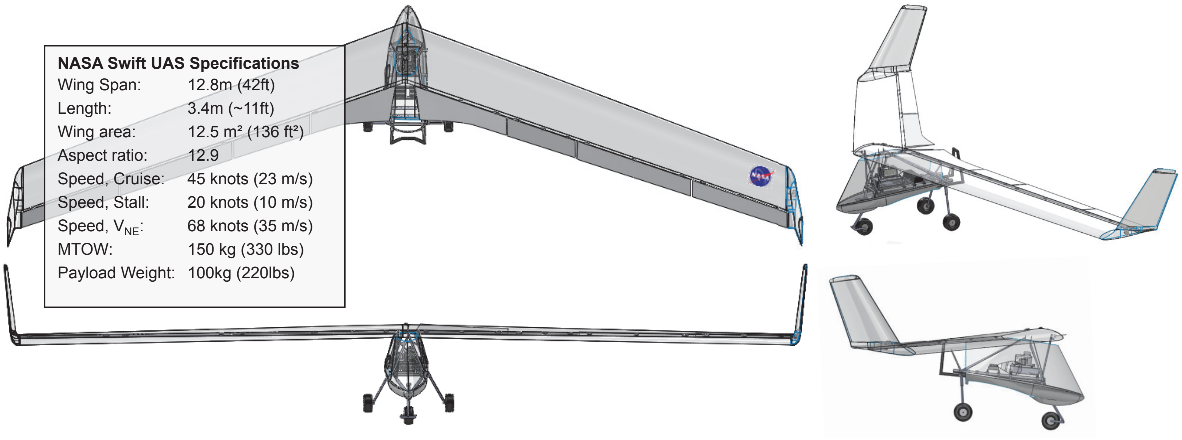
The Federal Aviation Administration (FAA) continues to formulate and refine its understanding of the safety aspects of integrating Unmanned Aircraft Systems (UAS) into the National Airspace System (NAS). The introduction of a UAS operation into controlled airspace is a modification of the airspace system, and therefore requires safety analysis to ensure that an acceptable level of safety is maintained. Consequently, the FAA performs a safety evaluation (of a UAS and its intended operations) as part of the process for approval of, and prior to issuing, the operating authority. However, this process occurs *a posteriori*, i.e., after the UAS has been developed.

There is a need for UAS developers to proactively consider safety aspects *during* system development, prior to the FAA safety evaluation, especially because safety is a key consideration in airworthiness. One of the primary goals of this work is to develop a methodology for such safety analysis, so as to support the derivation of an airworthiness statement and the subsequent application for the authority to operate in the NAS. We are also mainly concerned with small unmanned aircraft systems (sUAS), and the *Certificate of Authorization* (COA) process available to public entities, such as NASA or public universities, the process for which requires addressing system safety.[1,2] The process for civil UAS operators is different from the COA, requiring a *Special Airworthiness Certificate - Experimental Category* (SAC-Exp). Although this process provides a safety checklist, replacing previous requirements for a hazard analysis,[3] we believe that our approach is, nonetheless, also relevant.

Our rationale, here, is that early safety analysis will provide sUAS developers with a mechanism to (a) better understand and communicate safety considerations, in support of getting credit towards FAA goals (of determining that the UAS is safe to operate in the NAS), and (b) reduce the costs of a potential re-design based on the outcomes of a post-facto safety analysis. Early safety analysis, from the perspective of the developers, also provides a *bottom-up* approach to identify and address safety requirements arising from the need to mitigate potential hazards in the airspace where the UAS operations are intended.

This paper describes an integrated methodology for safety and systems engineering, which is exemplified by application to the Swift UAS[4] at the system level and subsequently at the aircraft subsystem level. In particular, we

[*] Senior Computer Scientist, Intelligent Systems Division. AIAA Member.
[†] Research Scientist, Intelligent Systems Division. AIAA Member.
[‡] Research Scientist, Intelligent Systems Division. AIAA Member.
[§] Robotics Researcher, Carnegie Mellon University.



NASA Swift UAS Specifications

Wing Span:	12.8m (42ft)
Length:	3.4m (~11ft)
Wing area:	12.5 m² (136 ft²)
Aspect ratio:	12.9
Speed, Cruise:	45 knots (23 m/s)
Speed, Stall:	20 knots (10 m/s)
Speed, V_{NE}:	68 knots (35 m/s)
MTOW:	150 kg (330 lbs)
Payload Weight:	100kg (220lbs)

Figure 1. Swift unmanned aircraft (UA) with some specifications and dimensions.

have considered a payload data subsystem, which has been designed to be re-used in other unmanned aircraft and missions at NASA Ames.

The rest of the paper is organized as follows: Section II provides a description of the Swift UAS and the payload subsystem, providing the application context in which we apply our integrated methodology (Section III). Section IV gives a detailed description of how the methodology was applied, exemplifying the steps in the processes and provides excerpts of the outcomes, i.e., the artifacts produced. Section V reflects on the application of our integrated methodology, and concludes the paper identifying avenues for future work.

II. System Description

A. Aircraft

The Swift UAS is a low-cost, experimental, all-electric unmanned autonomous vehicle system, being developed to support research goals in aeronautics and earth science. It comprises the unmanned electric Swift aircraft, dual redundant ground control stations (GCS) and communication links (900 MHz for telemetry and commands from the GCS, and 2.4 GHz for direct pilot commands) respectively. For this paper, we have mainly focused on the aircraft (Figure 1) and its subsystems, in particular, the Common Payload Data System (CPDS).

Both flight and ground software are implemented using the *Reflection* architecture,[5] a multi-component, event-driven, configurable software framework running on the Windows XP Embedded operating system.

B. Concept of Operations

The vehicle is designed to perform intelligent remote survey missions (e.g., as illustrated in Figure 2), fielding scientific instrument payloads over a remote location near phenomena of interest, processing this information in real-time, and autonomously adjusting aircraft behavior to maximize data return while maintain the safety of the aircraft. Processed mission data and control information is sent in real-time to the ground crew and remote observers, who can also provide real-time input as the mission proceeds.

C. Common Payload Data System

The Common Payload Data System (CPDS), shown in Figure 3, is a reusable payload data and flight control system. The CPDS is part of the Swift UAS avionics assembly, and contains the Command and Data Handling Unit (CDHU) sub-assembly. The CPDS provides flight data and air data sensing functionality, command processing, fault-tolerant data storage, wireless communication, and supports science payloads for a particular mission.

The CPDS has been designed to be reusable: it will be installed both in the Swift UAS (which is the focus of this paper), as well as in the NASA SIERRA UAS (out of scope for this paper). In the Swift UAS, the CPDS functions as the main flight computer, whereas in the SIERRA UAS its functionality is to process payload data as well to interface

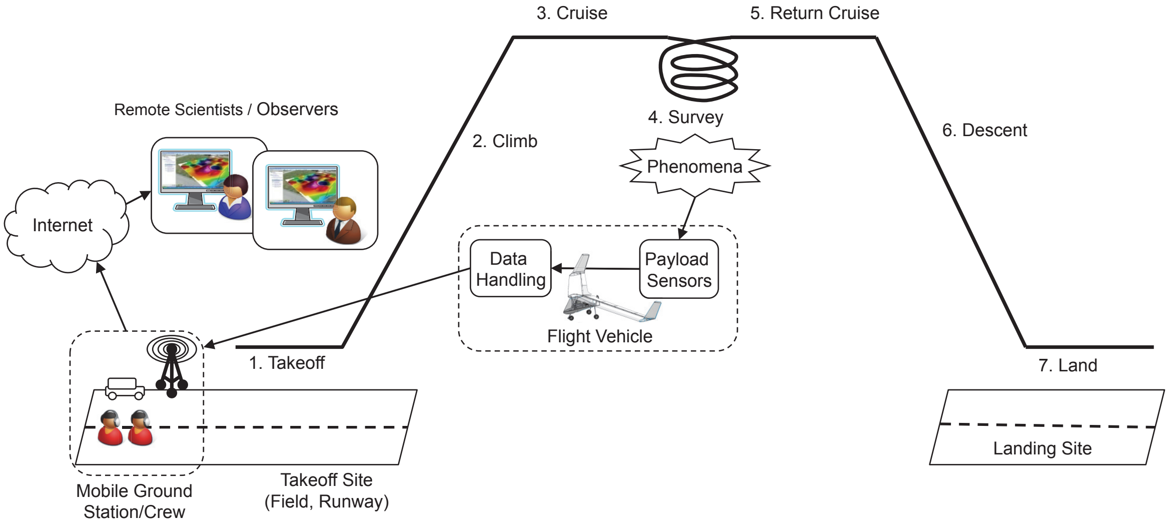


Figure 2. Swift UAS remote survey mission and data-flow concept.

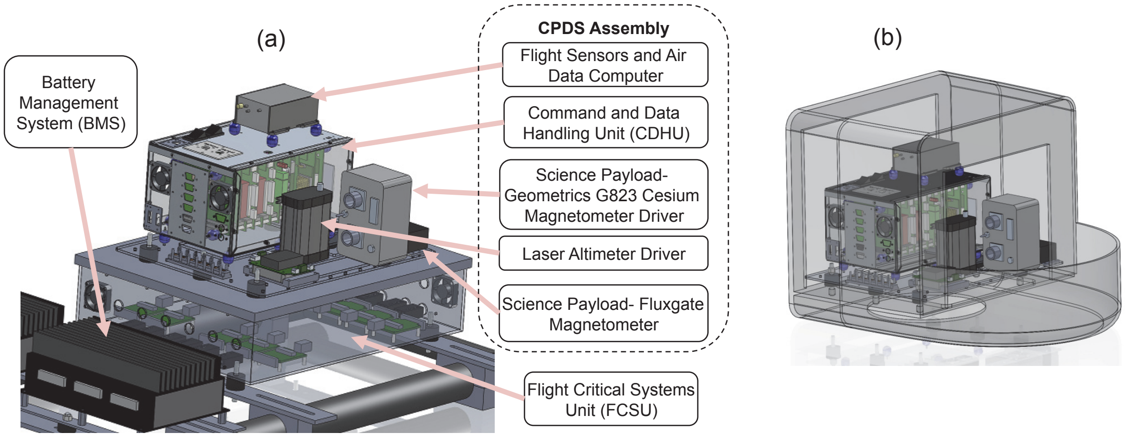

(a)
(b)
Battery Management System (BMS)
CPDS Assembly
Flight Sensors and Air Data Computer
Command and Data Handling Unit (CDHU)
Science Payload- Geometrics G823 Cesium Magnetometer Driver
Laser Altimeter Driver
Science Payload- Fluxgate Magnetometer
Flight Critical Systems Unit (FCSU)

Figure 3. Swift UAS CPDS assembly (a) CPDS and the Swift UAS avionics assembly (b) CPDS installed in the SIERRA UAS nose.

with the SIERRA Piccolo autopilot[a], in addition to the functionality described above.

III. Methodology

A. Systems Engineering Process

The approach to systems engineering being used to develop the Swift UAS is required to satisfy:
(i) NASA engineering process and procedural requirements for systems engineering[6] and software engineering,[7]
(ii) the requirements of the Airworthiness and Flight Safety Review Board (AFSRB), and
(iii) the various other relevant review boards in the agency.

The recommended two-phase process (Figure 4), features seven distinct sub-phases similar to a waterfall life-cycle approach. The project must pass through *gates*, i.e., key decision points and reviews, to proceed from one phase to the next. Phases methodically lead from concept studies to concept and technology development, design, fabrication, assembly, integration and testing (I&T), operations, and closeout.

However, this recommended lifecycle was determined to be infeasible given the constraint of limited resources: the Swift UAS engineering team comprised a very small group of engineers giving time to the project as they became

[a] Available from Cloud Cap Technology, Inc., `http://www.cloudcaptech.com/piccolo_system.shtm`

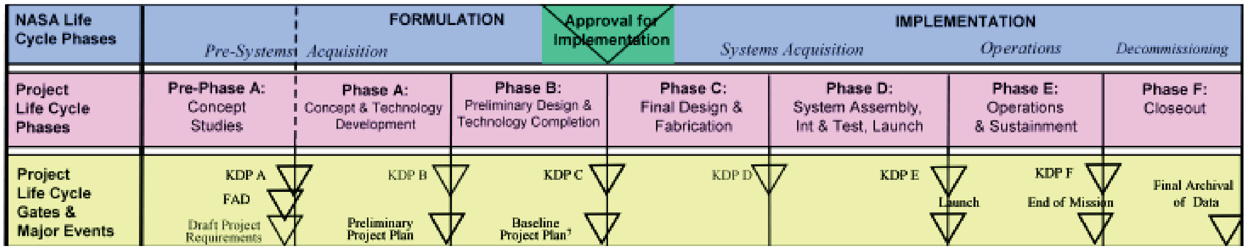

NASA Life Cycle Phases
FORMULATION
Approval for Implementation
IMPLEMENTATION
Pre-Systems Acquisition
Systems Acquisition
Operations
Decommissioning
Project Life Cycle Phases
Pre-Phase A: Concept Studies
Phase A: Concept & Technology Development
Phase B: Preliminary Design & Technology Completion
Phase C: Final Design & Fabrication
Phase D: System Assembly, Int & Test, Launch
Phase E: Operations & Sustainment
Phase F: Closeout
Project Life Cycle Gates & Major Events
KDP A
FAD
Draft Project Requirements
KDP B
Preliminary Project Plan
KDP C
Baseline Project Plan
KDP D
KDP E
Launch
KDP F
End of Mission
Final Archival of Data

Figure 4. NASA systems engineering lifecycle phases.[6]

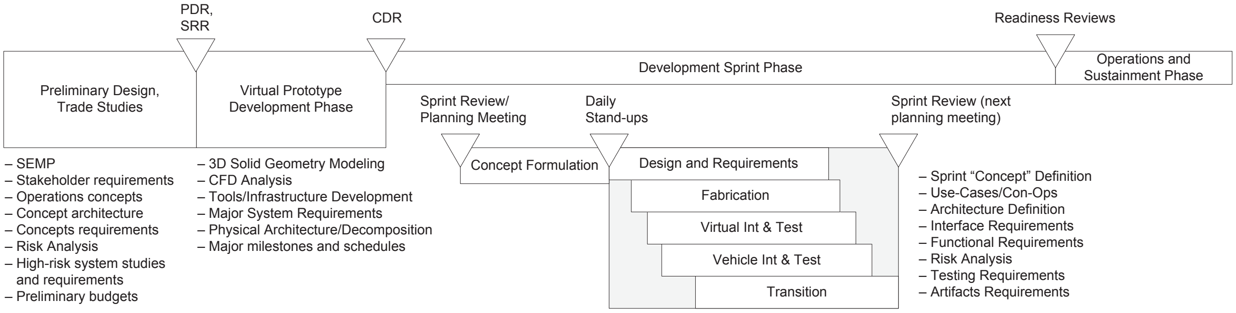

PDR, SRR
CDR
Readiness Reviews
Preliminary Design, Trade Studies
Virtual Prototype Development Phase
Development Sprint Phase
Operations and Sustainment Phase
Sprint Review/ Planning Meeting
Daily Stand-ups
Sprint Review (next planning meeting)
Concept Formulation
Design and Requirements
Fabrication
Virtual Int & Test
Vehicle Int & Test
Transition
– SEMP
– Stakeholder requirements
– Operations concepts
– Concept architecture
– Concepts requirements
– Risk Analysis
– High-risk system studies and requirements
– Preliminary budgets
– 3D Solid Geometry Modeling
– CFD Analysis
– Tools/Infrastructure Development
– Major System Requirements
– Physical Architecture/Decomposition
– Major milestones and schedules
– Sprint "Concept" Definition
– Use-Cases/Con-Ops
– Architecture Definition
– Interface Requirements
– Functional Requirements
– Risk Analysis
– Testing Requirements
– Artifacts Requirements

Figure 5. Agile process followed by the Swift UAS development team.

available, with two or three working on the project at any given time. Additionally, the recommended process requires a thorough investment in upfront design work and analysis that the engineering team might not have been able to provide, given the uncertain, transient and limited availability of the team members. However, the major artifacts, reviews, and documentation required by the NASA Procedural Requirements (NPRs) still apply.

B. Swift UAS Development

1. *Preliminary Design*

The development team adopted an experimental engineering process methodology inspired by the *agile* software process[8] (outlined in Figure 5). The process involves a limited up-front design process, as well as a preliminary hazard and risk analysis (described in more detail subsequently in this paper), which is the link to the safety methodology. The phase exits with a preliminary design review (PDR) and a system requirements review (SRR).

2. *Virtual Prototyping*

The second phase involves immediate development of end-to-end virtual environments. The goal of this phase is to develop an environment that has complete end-to-end functionality with limited depth, providing a "mile-wide, inch-deep" view of the system. At the end of this phase, the virtual environments are largely composed of *stubs*, i.e., empty implementation structures which meet interface requirements needed to stand in for a component without providing any implementation to meet functional requirements. Stubs also help engineers define the major architecture components and interface requirements between the components.

3. *Development Sprints*

After the virtual environments are created, the project enters the development sprint phase, in which the project spends most of its time. During this phase, stubs are replaced by actual implementation, adding depth to the project until all interface requirements and functional requirements are satisfied. The development sprint phase is composed of recursive *development sprints* which incorporates aspects of Phases A through D from the NASA systems engineering lifecycle (Figure 4), distributing the work performed in these phases throughout the project lifecycle.

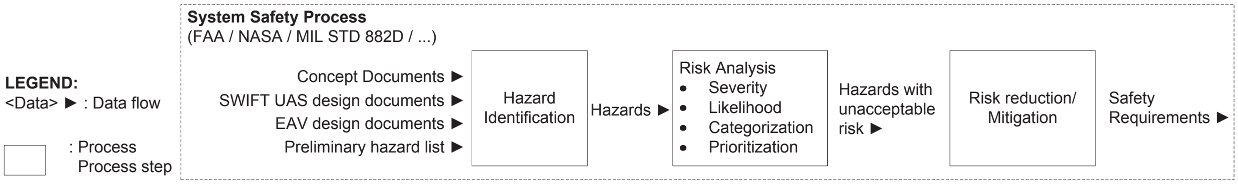


Figure 6. Safety process applied to the Swift UAS.

Each sprint is two to three weeks in duration, and is initiated by planning sessions to identify the sprint goals. This is followed by a development phase where engineers concurrently design, develop, integrate, and test. At the outcome of each sprint, engineers are required to have realized the sprint concept while generating fragments of the required NPR artifacts (e.g., additional requirements, risk analysis artifacts, testing requirements, and test results).

A sprint results in a complete end-to-end system. The complete set of tests identified at that point in development represents the regression test suite. The team utilized web-based collaboration tools extensively to coordinate development, recording updates to online weblogs, utilizing wiki pages, utilizing custom databases through Microsoft Sharepoint[b], and controlling documents and their revisions through the Subversion version control system[c].

C. Safety Methodology

Our process for system safety (Figure 6) is derived from the framework of a safety risk management plan, e.g., as recommended in MIL-STD-882D,[9] the NASA Facility System Safety Guidebook,[10] or the FAA System Safety Handbook.[11] We include safety considerations into system design at an early stage through the systematic identification of hazards, risk analysis and risk management. These activities are intertwined, and phased, with system development, i.e., they are applied at the early stages of system development and iteratively as system development progresses (to the subsystem and component levels).

1. Hazard Identification

The safety methodology begins with hazard identification. Applied at the early stages, this forms part of the preliminary hazard analysis (PHA). The input to this activity is a preliminary hazard list (PHL), the available and relevant design documentation for the Swift UAS and its predecessor vehicle, i.e., the Exploration Aerial Vehicle (EAV), and the concept of operations (ConOps) document. During hazard identification, we apply techniques such as

Functional Hazard Assessment (FHA): A systematic, top-down method where system functions and their associated failure conditions are listed, together with their effects, classification of severity, likelihood of occurrence and mitigation measures (which are, eventually stated as requirements). The focus here, is on the functional system breakdown, rather than other decompositions of the system, e.g., physical breakdown.

Failure Modes and Effects Analysis (FMEA): A bottom-up approach for hazard identification by enumerating the failure modes, and corresponding effects, based on the physical and functional breakdown of the system. The complexity of this analysis was managed by restricting the identification of failure modes to the subsystems identified in the physical breakdown of the Swift UAS.

A suite of other techniques can be applied, such as fault tree analysis (FTA), and event-tree analysis (ETA), which allows reasoning, respectively, about the failure paths and event chains leading to system hazards. For this paper, however, we mainly focused on FHA and FMEA due to resource limitations.

2. Risk Analysis and Risk Reduction/Mitigation

Once hazards have been identified, we systematically categorize them based on their risk i.e., consequence severity (Table 1) and the respective likelihood of occurrence (Table 2). The combination of the severity and likelihood, gives a risk categorization matrix (Table 3), containing hazard risk indices. A grouping of risk indices (Table 3) gives a risk

[b] http://sharepoint.microsoft.com/
[c] http://subversion.tigris.org/

Table 1. Hazard severity levels

Level	Meaning	Description	*Health, Safety, Environment*	*Mission Success*
5	NO SAFETY EFFECT	No effect to safety of the aircraft.	No effect	Halt of mission execution, loss of operational time conducting mission.
4	MINOR	Aircraft may lose redunant system. Flight control authority impared.	No effect	Likely mission failure, mission may be resumed
3	MAJOR	Aircraft may lose systems beyond redundant system. Flght control authority impaired. Flight termination may be required.	No effect	Likely mission failure, mission may be resumed
2	HAZARDOUS	Aircraft may lose control. Flight termination and loss of aircraft a likely outcome.	Aircraft may have to terminate, slight increased safety risk.	Certain mission failure
1	CATASTROPHIC	Loss of aircraft and control.	Uncontrolled loss of aircraft, increase in safety risk.	Certain mission failure

Table 2. Hazard occurrence probability levels

Level	Description	Definition
E	Extremely Improbable	p < 1E-09
D	Extremely Remote	1E-09 < p < 1E-07
C	Remote	1E-07 < p < 1E-06
B	Somewhat Probable	1E-06 < p < 1E-05
A	Probable	p > 1E-05

category, which is tied to a specific risk reduction/mitigation decision, i.e., elimination, control or acceptance. We have defined four categories as given in Table 4, together with the risk reduction decisions.

Thereafter, we define safety requirements so as to mitigate the identified hazards, with the safety requirements being defined based on a hazard reduction precedence,[10] i.e., first design to eliminate, else design to control, the hazard. If neither is possible, then provide safety devices, followed by providing warning devices. If these are also not possible, provide special procedures or training. The intent of this reduction precedence is to reduce risk to an acceptable level.

Table 3. Risk Categorization Matrix

	SEVERITY	CATAS-TROPHIC	HAZARDOUS	MAJOR	MINOR	NO SAFETY EFFECT
LIKELIHOOD	Index	*1*	*2*	*3*	*4*	*5*
PROBABLE	*A*	1A	2A	3A	4A	5A
SOMEWHAT PROBABLE	*B*	1B	2B	3B	4B	5B
REMOTE	*C*	1C	2C	3C	4C	5C
EXTREMELY REMOTE	*D*	1D	2D	3D	4D	5D
EXTREMELY IMPROBABLE	*E*	1E	2E	3E	4E	5E

3. *Range Safety*

One of the primary mechanisms for ensuring safe UAS *operations* is *range safety*. In general, this is the application of safety policies, principles and techniques to protect people and property from hazards arising from UAS operations. Range safety has been quantified using the *Joint Advanced Range Safety Mission Planning Software* (the JARSS tool)[12] which can estimate casualties (in terms of expected casualties), and impact dispersions, based on population and atmospheric data for the mission range.

The safety process (Figure 6) complements, and is compatible with, range safety. Specifically, range safety considers the hazards in the airspace, i.e., *the range*, where operations are conducted, and gives a quantitative basis for the risk indices that they will be assigned. These identified hazards are included into the PHL, following which the safety process can be applied to define the relevant safety requirements.

Table 4. Risk categories and risk reduction decisions

Category	Hazard Risk Indices	Decision
Highest Risk	1A	Must mitigate
Moderately High Risk	2A, 3A, 1B, 2B, 1C	Must mitigate
Moderate Risk	3A, 3B, 4B, 2C, 3C, 1D, 2D, 1E, 2E	Must monitor
Low Risk	5A, 5B, 4C, 5C, 3D, 4D, 5D, 3E, 4E, 5E	No action required

D. Linking Safety and System Development

We consider the data generated from the systems development and the safety methodology to integrate the two processes. As described earlier, the preliminary design phase (Figure 5) includes activities to identify and document stakeholder requirements, and requirements arising from the concept of operations, which are documented as a set of tables. In this phase, a concept architecture is also created together with a functional and physical breakdown; these are also documented, in part, in a tabular form.

The hazard identification and analysis steps (Figure 6) generate safety requirements, depending on the level at which they are applied, e.g., system safety requirements are obtained through a system-level preliminary hazard analysis, whereas lower-level safety requirements are obtained through a subsystem hazard analysis. We maintain the outcomes of each step in the process in a set of related tables. These form a concrete interface to the requirements table, i.e., the safety process is used to define the safety requirements that are effectively a subset of the set of Swift UAS requirements to be implemented during system development. In particular, some of the safety requirements are related to functional safety, and therefore also related to the functional requirements to be implemented by the system design. The remainder of the safety requirements are addressed as part of the development and lifecycle activities, e.g., airframe and hardware design, operating procedures, maintenance, etc.

The tables also concretely trace the hazards identified from the safety analysis to safety requirements and eventually to the verification procedures, e.g., system tests, required as the evidence of safe design and operation, in the statement of airworthiness. In effect, the safety methodology is applied at each step during development and at each level of the system iteratively; the resulting safety requirements and mitigation measures form part of the input required for system development both for the subsequent development steps and the lower levels of the system. Safety analysis links with system development mainly when defining (system, subsystem or component) requirements or when a design is modified. The latter is required to ensure, via re-analysis, that either design modifications do not violate system safety requirements, or that safety requirements need to be appropriately altered.

IV. Exemplification

In this section, we exemplify the application of the methodology by giving excerpts of artifacts produced by the integrated system development and safety methodologies. As identified in the previous section, the rapid development of functionality for the avionics and flight-critical systems, which occurs in two to three week sprints, is noteworthy. This required the safety process to be tailored such that it would keep up with the development and actively influence it. Additionally, given the small development team, resources constraints were significant.

Thus, safety analysis was mainly performed in the preliminary design phase and then subsequently during the development sprint phases. The virtual prototype development phase was used to refine the understanding of system behavior in the context of the identified system level hazards, and to guide the subsystem hazard analysis. We did not apply the safety methodology in this step, due to resource limitations.

To facilitate collaboration, most documents (including reports, design documents, studies and engineering drawings) were printed and arranged on the walls of the lab, organized by subsystem. Engineers could quickly find subsystem drawings during standup meetings and make red-line changes to system drawings. If approved, the changes were committed and added to document control.

A. System Level – Aircraft and Avionics

1. Preliminary Design

The development team followed the methodology described in Section III: as the first step, they completed a preliminary design and initial studies to identify the overall architecture, and develop the high level aspects of the Systems

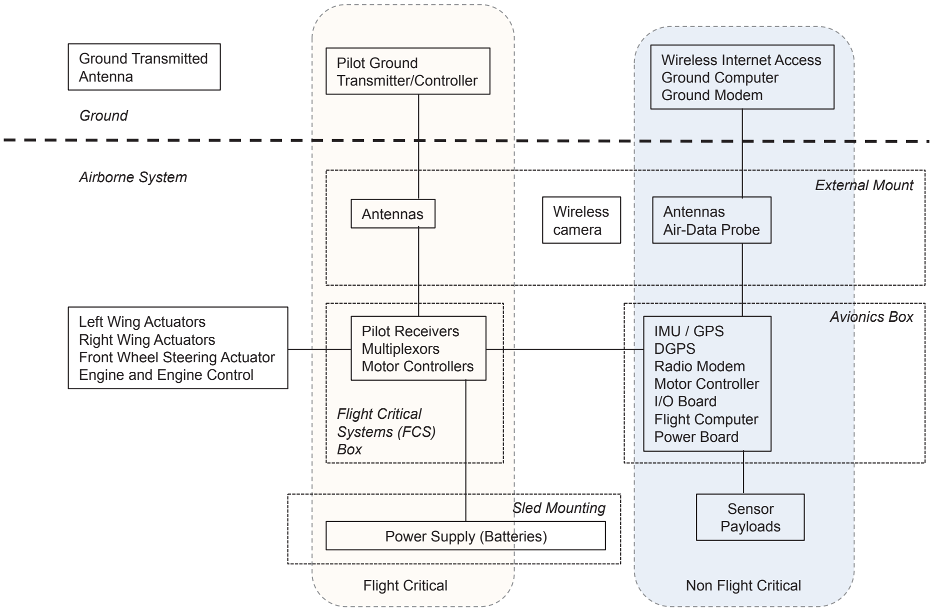


Figure 7. Swift UAS onboard avionics architecture concept highlighting flight critical and non-flight-critical components.

Table 5. Preliminary hazard list

Subsystem	Hazards
Aircraft (UA)	Cable failure Connector failure Software/ firmware/ Avionics CPU errors Loss of communication Mechanical fastener failure Stuck servo Component failure Fire hazard from overheating

Engineering Management Plan (SEMP). This included stakeholder requirements, concept of operations, conceptual architectures, logical/physical/functional architecture definitions, and high level requirements.

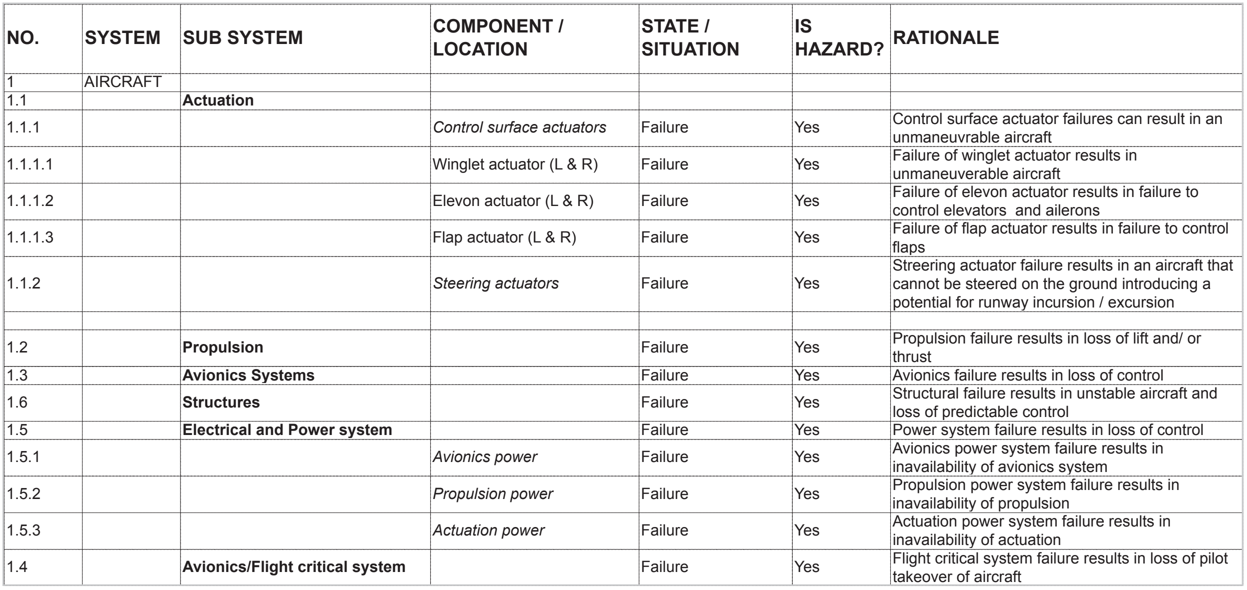
For example, Figure 7 gives a high-level overview of the onboard avionics architecture concept highlighting the subsystems and components involved in the the airborne system, and their connections to the ground system. The figure also highlights the grouping of the flight-critical and non flight critical components, which cuts across the boundaries of the airborne and ground systems respectively. The avionics concept is designed to meet the system and functional requirements and also represents a physical breakdown of the system.

2. *Hazard Analysis*

Safety analysis is performed in parallel with system design, starting with hazard identification. Table 5 shows a preliminary list of hazards identified from a predecessor vehicle of the Swift UAS, the Exploration Aerial Vehicle (EAV). Table 6 shows a fragment of an additional list of hazards identified from the concept and design documents; this list enumerates the failure hazards based on the functions to be provided and subsequently based on a physical breakdown of the system, e.g., the actuation system and the electrical and power system, have been expanded in Table 6, to show failure hazards of the components.

For the Swift UAS, each required function was allocated to an item in the physical architecture. Consequently, the functional breakdown shows a one-to-one mapping with the physical system breakdown. A fragment of the hazard analysis based on the functional/physical breakdown at the system level, i.e., of the avionics concept architecture (Figure 7), is given in Table 7. The table shows failure conditions during a specific operating phase, i.e., descent, and a specific sub-phase, i.e., approach. The rationale here, is that not all failures are hazards. Rather, hazards are a state or set of system conditions that, together with other conditions in the environment, will lead to a loss event.[13] For each

Table 6. Fragment of additional hazard list



NO.	SYSTEM	SUB SYSTEM	COMPONENT / LOCATION	STATE / SITUATION	IS HAZARD?	RATIONALE
1	AIRCRAFT					
1.1		**Actuation**				
1.1.1			*Control surface actuators*	Failure	Yes	Control surface actuator failures can result in an unmaneuvrable aircraft
1.1.1.1			Winglet actuator (L & R)	Failure	Yes	Failure of winglet actuator results in unmaneuverable aircraft
1.1.1.2			Elevon actuator (L & R)	Failure	Yes	Failure of elevon actuator results in failure to control elevators and ailerons
1.1.1.3			Flap actuator (L & R)	Failure	Yes	Failure of flap actuator results in failure to control flaps
1.1.2			*Steering actuators*	Failure	Yes	Streering actuator failure results in an aircraft that cannot be steered on the ground introducing a potential for runway incursion / excursion
1.2		**Propulsion**		Failure	Yes	Propulsion failure results in loss of lift and/ or thrust
1.3		**Avionics Systems**		Failure	Yes	Avionics failure results in loss of control
1.6		**Structures**		Failure	Yes	Structural failure results in unstable aircraft and loss of predictable control
1.5		**Electrical and Power system**		Failure	Yes	Power system failure results in loss of control
1.5.1			*Avionics power*	Failure	Yes	Avionics power system failure results in inavailability of avionics system
1.5.2			*Propulsion power*	Failure	Yes	Propulsion power system failure results in inavailability of propulsion
1.5.3			*Actuation power*	Failure	Yes	Actuation power system failure results in inavailability of actuation
1.4		**Avionics/Flight critical system**		Failure	Yes	Flight critical system failure results in loss of pilot takeover of aircraft

condition, we describe the effect on the system, and identify the risk index based on the likelihood and severity of the hazard (as described in Section III). The table also shows the corresponding mitigation mechanisms.

Table 8 shows how the (highlighted) requirements derived from hazard analysis[d] are integrated as part of the overall requirements of the Swift UAS. The non-highlighted requirements are the functional requirements that drive the preliminary design. In addition, Table 8 shows the linked allocation items and verification methods/allocations. In this way, there exist traces from the outcomes of the safety methodology, i.e., safety requirements, to design artifacts, i.e., system requirements, verification allocations and implementation allocations.

3. *Contingency Management*

The hazard analysis, performed as given in Table 7, and range safety analysis, are used to ensure that the aircraft will not stray outside the mission boundary. These impose requirements such as the need for a capability for direct pilot intervention as required, or a flight termination system. For example, a mitigation measure shown in Table 7 is "ground pilot intervention", and is a risk mitigation protocol in the Swift UAS contingency management system (CMS).

Three CMS have been considered, to be applied in the following order of precedence:

(1) If there is a failure of the primary pilot system, then go to the secondary system, which is redundant and on a different channel.

(2) If that fails, engage the onboard autopilot.

(3) If that too fails, force a spiral descent to impact.

B. Swift UAS Virtual Prototype Development

After passing a preliminary design review and receiving project approval, the system progressed into the virtual prototype development phase. The development team relied heavily on virtual I&T environments developed in SolidWorks[e] and Reflection (Section II)

The rapidly-reconfigurable component-based nature of Reflection provided a mechanism for automated regression testing of software and hardware in isolation, in small assemblies, and for the fully integrated flight vehicle with hardware in the loop simulation. During the virtual prototyping phase, the project developed two complete virtual end-to-end system models: The first model was a complete solid geometry model in SolidWorks. The team generated a high resolution 3D model of the airframe using a robotic arm device, translated this model into SolidWorks, and

[d] Also indicated by the source column, where the prefix "HR" indicates that the source is the hazard analysis.

[e] `http://www.solidworks.com/`

Table 7. **Hazard analysis fragment for the Swift UAS.**

ID	HAZARD DESCRIPTION	EFFECT ON SYSTEM	LIKELIHOOD	SEVERITY	RISK INDEX	MITIGATION
OPERATING PHASE	Descent (DE)					
SUB-PHASE	Approach (APP)					
	ACTUATION					
PHA_DE.APP_ACT_002	Elevon actuator failure	(1) Loss of control of flight surface controlled by Elevon actuator (2) Aircraft stall.	Remote	Hazardous	2C	Preflight inspection
PHA_DE.APP_ACT_003	Flap actuator failure	(1) Loss of control of flaps (2) Aircraft stall	Remote	Hazardous	2C	Preflight inspection
PHA_DE.APP_ACT_004	Front wheel steering actuator failure	No known consequence during approach sub-phase	Remote	Hazardous	2C	Preflight inspection
	AVIONICS HARDWARE: SENSORS					
PHA_DE.APP_AVCS_001	IMU/GPS (Rockwell Collins Athena 111m) Failure	(1) Loss of GPS signal (2) Incorrect waypoint and heading data supplied to autopilot (3) Drift outside range safety area	Extremely Remote	Hazardous	2D	(1) Failsafe autopilot forces controlled sprial to ground (2) Ground station pilot controller overrides autopilot
PHA_DE.APP_AVCS_002	DGPS (Novatel OEM4-G2) Failure		Extremely Remote	Hazardous	2D	
PHA_DE.APP_AVCS_003	Air data probe Failure	(1) Incorrect airdata supplied to autopilot (2) Overspeed (3) Aircraft stall (4) Loss of flight	Remote	Hazardous	2C	
	AVIONICS SOFTWARE: AUTOPILOT					
PHA_DE.APP_AVCS_012	Flight management system (FMS) Failure	(1) Incorrect computation of control surface signals (2) Incorrect actuator signals supplied (3) Loss of control over flight surface (4) Loss of mission (5) Loss of flight (6) Incorrect computation of waypoints and headings	Remote	Major	3C	(1) Ground station pilot controller overrides autopilot (2) Failsafe autopilot intervenes when failure of autopilot detected
PHA_DE.APP_AVCS_013	AP Failure		Remote	Hazardous	2C	
PHA_DE.APP_AVCS_014	Waypoint data Failure		Remote	Hazardous	2C	
PHA_DE.APP_AVCS_015	Autopilot module		Remote	Hazardous	2C	

added only the major internal structures. All details and subsystems were stubs, modeled as rectangular blocks. The second virtual model generated was an end-to-end virtual embedded simulation environment in Reflection.

An existing UAS component-based configuration was duplicated as the initial baseline model, which included a flight dynamics module, actuator dynamics modules, sensor hardware emulators, a flight management system, an autopilot control system, command processing module, and data communication module.

Aerodynamics models were generated by analyzing the solid geometry in a Navier-Stokes Computational Fluid Dynamics (CFD) solver to generate the aircraft flight dynamic derivatives and aerodynamic coefficient tables. Additional vortex-lattice analyses were performed to verify the CFD model. Rendering models of the Swift UAS were generated, and actuators adjusted to reflect the aircraft configuration.

Thereafter, several stubs of components were added, such as a control allocation block that translates the autopilot output to the unique control surface configuration of the Swift UAS. Finally, an existing UAS hardware ground station was utilized as the baseline, which included visualization components, hardware RF modem, and hardware pilot controls. The major system architecture was finalized during this phase, major project milestones identified, and a critical design review (CDR) was conducted to allow the process to progress to the development sprint phases.

C. Subsystem Level – Common Payload Data System

1. *Concept Formulation*

A request for payload data processing was received by the Swift UAS development team in the middle of a development sprint cycle. The team responded by performing a two week development sprint to perform a systems engineering analysis, which included identifying stakeholder requirements and generating a concept of operation (ConOps) that stakeholders could approve. This corresponds to the *Concept Formulation* and *Design and Requirements* steps in the

Table 8. Requirements fragment for the Swift UAS including safety requirements (highlighted).

Req ID	Requirement Name	Source	Implementation Allocation	Verification Method/ Allocation
RF.1.0	Provide the complete Swift UAS system solution (ground, air, procedures, etc.).	PA.1.0		
RF.1.1	Provide a flight vehicle system (FVS).	PA.1.1		
RF.1.1.1	(AVS) Provide onboard avionics system (AVS) for sensing, control, automation	PA.1.1.1		
RF.1.1.1.1	Provide flight critical control system (FCS)	PA.1.1.1.1		
RF.1.1.1.2	System must provide control and data handling system (C&DH)	PA.1.1.1.2		
RF.1.1.1.2.1	CDHU Hardware	PA.1.1.1.2.1		
RF.1.1.1.2.2	FMS (Flight Management System)	PA.1.1.1.2.2		
RF.1.1.1.2.3	ESS (Embedded Software Systems)	PA.1.1.1.2.3		
RF.1.1.1.2.4	AP (Autopilot System)	PA.1.1.1.2.4		
RF.1.1.1.3	System must communicate with the ground and external systems (COM).	PA.1.1.1.3		
RF.1.1.1.4	System must be able to sense its current state and surrounding environment (SENS)	PA.1.1.1.4		
RF.1.1.2	(PYLD) System must manage mission payloads and experiments (PYLD)	PA.1.1.2		
RF.1.1.4	(PRLP) Provide propulsion (PRLP)	PA.1.1.4		
RF.1.1.4.1	Provide an electric propulsion system.	PA1.1.4.1, RS.1.1.2		
RF.1.1.4.1.1	Engine must be controllable by remote pilot and by CPU/autopilot system.	RS.1.1.1, RS.1.4.1	Engine Systems	Pre-flight checklist
RF.1.1.4.1.2	CPU/autopilot system must be able to monitor engine and motor controller temperature.	HR.1.3.1	Engine Systems	Pre-flight checklist
RF.1.1.4.1.3	Must have engine current and voltage monitoring during flight	HR.1.3.3, HR.1.3.2	Engine Systems	Pre-flight checklist
RF.1.1.4.1.4	Desirable to have engine RPM feedback	HR.1.3.3, HR.1.3.2	Engine Systems	N/A (requirement closed)
RF.1.1.4.1.5	Must have preflight inspection of prop in checklists.	HR.1.3.4, HR.1.3.5	Pre-flight checklist	Pre-flight checklist
RF.1.1.4.1.6	Must inspect runway for FOD before flight operations can be conducted.	HR.1.3.4, HR.1.3.5	Pre-flight checklist	Pre-flight checklist
RF.1.1.4.1.7	Engine mounting (assembly and construction) will be performed by airframe manufacturer.	HR.1.3.6	Engine Systems	Engine integration flight test.
RF.1.1.4.1.8	Engine mount will be inspected after final flight.	HR.1.3.6	Engine Systems	Post-flight checklist
RF.1.1.4.1.9	Engine software will be checked during pre-deployment checkout.	HR.1.3.7	Pre-deployment checklist	Pre-deployment checklist

Table 9. Requirements excerpt for the CPDS.

Req ID	Requirement Name
RF.1.1.2.1	(CPDS) - Common CDHU & Payload Data System assembly requirements
RF.1.1.2.1.1	CPDS assembly must include the CDHU, mission payload sensors, antennas, hardware assembly, mounting hardware and design.
RF.1.1.2.1.1.1	CPDS assembly must support all assigned subsystems (electrical, communication, structural, environmental, etc.)
RF.1.1.2.1.1.2	CPDS assembly must mount sensors in a way that allows for proper sensor operation (eg, altimeter downward facing, etc.)
RF.1.1.2.1.2	CPDS must log data from (record the data stream from) the Geometrics magnetometer, flux-gate magnetometer, and laser altimeter.
RF.1.1.2.1.3	CPDS must log data from (record the data stream from) the INS (Inertial Navigation System) system position and orientations
RF.1.1.2.1.4	CPDS assembly must be common (as much as possible) between both the Swift UAS and the Sierra UAS
RF.1.1.2.1.5	CPDS must synchronize with the GPS clock from UAS system's INS system clock
RF.1.1.2.1.6	CPDS must draw power from the the aircraft or from onboard power
RF.1.1.2.1.7	CPDS must be powerable from a lab power supply.
RF.1.1.2.1.8	CPDS mounting must provide mechanical isolation and protection from the UAS structure (vibration, shock)

development sprint phase (Figure 5).

The ConOps described the envisioned flight test process, identifying personnel needed and the procedure for collecting data during operations. Subsequently, requirements were derived from the ConOps and architecture, excerpts of which are shown in Table 9.

The team then identified necessary artifacts to be developed, such as procedural check lists and integration tests that needed to be performed. The plan was approved by stakeholders during a combined preliminary/critical design review.

2. *Failure Modes and Effects Analysis*

For the CPDS, i.e., aircraft subsystem level, we performed both subsystem hazard analysis (SSHA) and failure modes and effects analysis (FMEA). SSHA is performed in the same way as PHA, but with a narrower focus—in this case, the CPDS. FMEA, on the other hand, permitted a bottom-up way to trace from individual failure conditions and failure modes to system-level hazards and to the relevant safety requirements. For instance, Table 10 shows how specific CPDS component failure modes such as wire overheating trace to fire hazards at the system level and a corresponding system requirement (with ID RS.4.1). Similarly, other failure modes such as from vibration or impacts, trace to failure hazards at the subsystem level, i.e., sensor loss, and a corresponding CPDS requirement with ID RF.1.1.2.1.9.4 (Note that this requirement has not been shown in the requirements fragment of Table 9).

These motivated specialized lower level analyses on the CPDS components, e.g., stress analysis on the CPDS mounting plate, which was identified as a critical structural load component. Similarly, to control component and wire overheating failure modes (and thereby control a potential fire hazard), power requirements analysis and wire sizing analysis were used, respectively, to give the power supply specifications, and to select the appropriate wires sizes.

3. *Resulting Design*

Based on the concept formulation and the safety analysis, i.e., FMEA, a concept architecture was proposed for a common payload data system (CPDS), as shown in Figure 8. The figure highlights the reuse capability of the CPDS

Table 10. Failure modes and effects analysis (FMEA) fragment for the CPDS.

ID	FAILURE MODE	CAUSE	ASSOCIATED PA ID	EFFECT	MITIGATION	DERIVED SAFETY REQUIREMENT
HR.1.4.7	**Common Payload Data System**					
HR.1.4.7.1	Loss of single component	Component failure	PA.1.1.2.1	Loss of single payload sensor	Use components from reputable sources, CPDS is non-flight critical	RF.1.1.2.1.11
HR.1.4.7.2	Switch failure - turns off in flight	Vibration, shock, mechanical failure	PA.1.1.2.1	Loss of single payload sensor	Design to account for vibration, shock environment	RS.4.2
HR.1.4.7.3	Loss of power	Power system failure, connector failure	PA.1.1.2.1	Loss or partial loss of CPDS functionality	CPDS is non-flight critical	RF.1.1.2.1.11
HR.1.4.7.4	Component loss of power	Power system failure, connector failure	PA.1.1.2.1	Loss of single payload sensor	CPDS is non-flight critical	RF.1.1.2.1.11
HR.1.4.7.5	Loss of/intermittent signal (COM)	Connector failure, wire failure in vibration, shock	PA.1.1.2.1	Loss of single payload sensor	CPDS is non-flight critical	RF.1.1.2.1.11
HR.1.4.7.6	Connector failure	Poor electrical workmanship	PA.1.1.2.1	Loss of single payload sensor	Electrical work performed and inspected by certified personnel according to NASA STD 8739.	RS.4.1
HR.1.4.7.7	Damage during inspection	Accidental knocking	PA.1.1.2.1	Loss of single payload sensor	CPDS is non-flight critical	RF.1.1.2.1.11
HR.1.4.7.8	Overheating of components	Ambient temperature too high, insufficient cooling	PA.1.1.2.1	Potential fire	Monitoring of ambient temperature, design safety factor	RS.4.1
HR.1.4.7.9	Overheating of wires	Ambient temperature too high, insufficient cooling	PA.1.1.2.1	Potential fire	Monitoring of ambient temperature, design safety factor	RS.4.1
HR.1.4.7.10	Shorting/electrical	Poor electrical workmanship	PA.1.1.2.1	Potential fire	Electrical work performed and inspected by certified personnel according to NASA STD 8739.	RS.4.1
HR.1.4.7.11	Component comes loose of mount	Mounting failure, perhaps in vibration and shock	PA.1.1.2.1	Loss of single payload sensor, may cause mechanical damage to other components	Use 2 methods of fastening where possible	RF.1.1.2.1.9.4
HR.1.4.7.12	Wire damage	Vibration, improper handling	PA.1.1.2.1	Loss of single payload sensor	Wires shall be secured and routed according to NASA STD 8739.	RS.4.1
HR.1.4.7.13	Shock/impact effects/damage	Shock, vibration from UAS, or foreign objects	PA.1.1.2.1	Loss or partial loss of CPDS functionality	Vibration isolation mounts on CPDS plate. Enclosed components.	RF1.1.2.1.9.4
HR.1.4.7.14	Vibration effects/damage	Shock, vibration from UAS, or foreign objects	PA.1.1.2.1	Loss or partial loss of CPDS functionality	Vibration isolation mounts on CPDS plate. Enclosed components.	RF1.1.2.1.9.4

for use in the Swift UAS and the SIERRA UAS.

As shown in Figure 8, the CPDS includes the command and data handling unit (CDHU), mission payload sensors, antennas, hardware assembly, mounting hardware. The CPDS is mounted on a vibration isolated mounting plate that can be mounted in both the Swift UAS and the SIERRA UAS. This design choice is the consequence of a hazard reduction measure, i.e., design to control the hazard, to account for vibrations and shocks in the operational environment.

V. Discussion and Conclusion

Given the use of an experimental agile development process, the Swift UAS team required a process for managing and mitigating risks, so as to comply with applicable NASA NPRs and to be transparent to external review committees. The method for safety analysis presented in this paper, its integration into the overall development process, and the creation of an operationalization (e.g., step by step procedures to execute the process) for the engineers on the Swift UAS development team was a key enabling tool that enhanced overall productivity and effectiveness.

However, the lack of automated tools made this process cumbersome, a problem which was accentuated by the agile process; during development sprints, many artifacts such as requirements, hazard lists, tables of mitigations, etc. would be added, removed, or modified. Consequently, ensuring consistency between each of the artifacts was difficult, requiring engineers to search through all possibly effected tables and manually update. Tracing requirements and implementation through the risk tables often exposed inconsistencies that took time to address and correct. Further, the source control methodology used by the team did not allow concurrent editing of the tables, causing additional difficulties.

Automated web-based tools providing bilateral traceability, consistency, concurrent editing, and guides to walk engineers through the process would facilitate adoption of the methodology. In addition, tool support would also be useful from the perspective of safety assessment and to increase transparency in the assurance process, i.e., for the members of the review boards to review and understand the safety analysis data produced, how this data traces to high-level concerns and assures safe system design and safe operations. We are actively working on tool support to address these shortcomings.[14–17]

The COA application process is predicated on a worst case hazard analysis for generic UA for a specific class of airspace.[18] The COA, then, is effectively a means to demonstrate that these hazards have been suitably mitigated using emergency procedures, range safety, specific functions (such as automatic flight termination), and so forth. The links between the safety / hazard analysis performed by the FAA Air Traffic Organization (and possibly to be also performed by the proponent during UAS design), the mitigations required, the documentation supplied by the proponent, and the COA goal of demonstrating a level of safety equivalent to that of manned operations, is largely implicit; additionally,

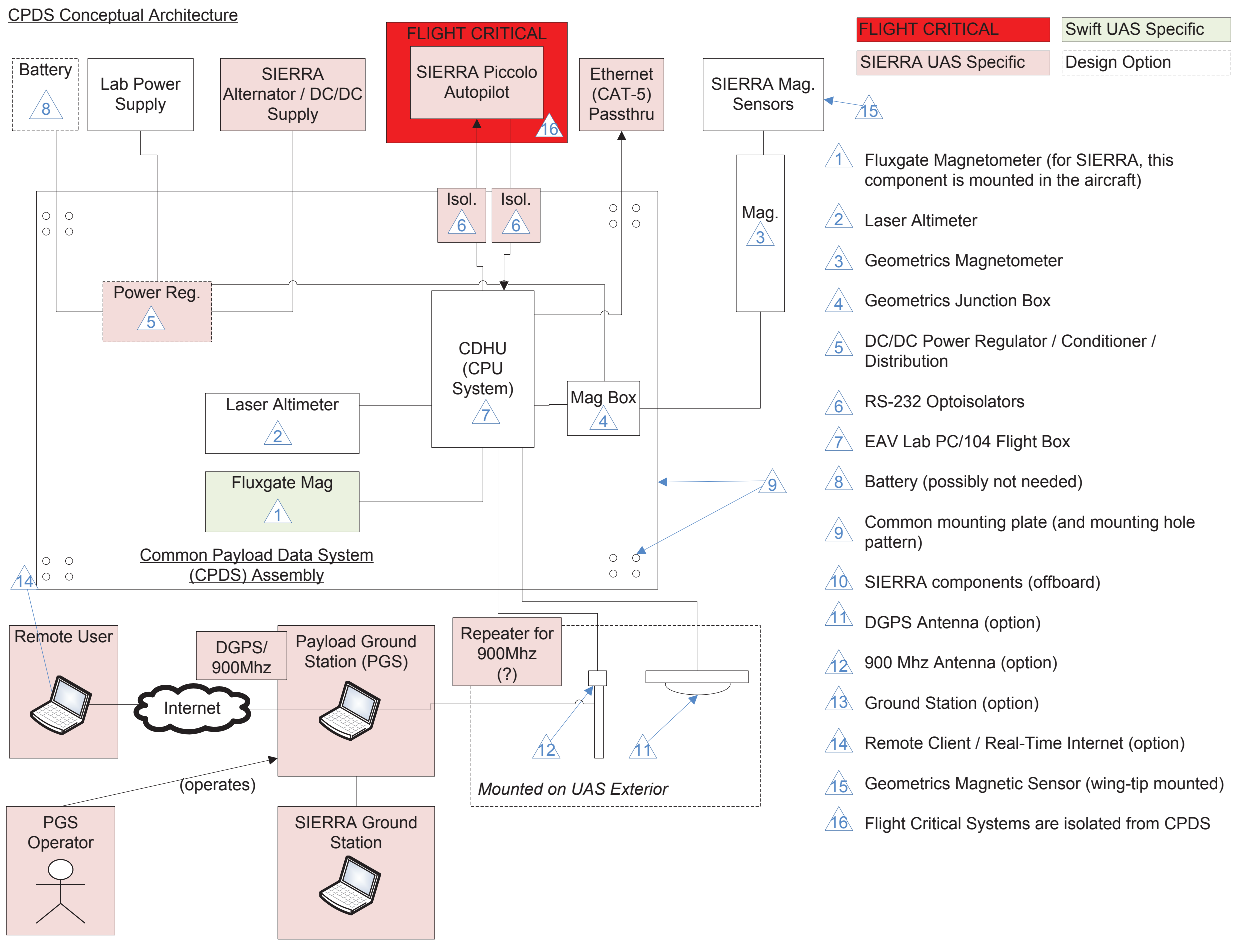

CPDS Conceptual Architecture
FLIGHT CRITICAL
Swift UAS Specific
SIERRA UAS Specific
Design Option
Battery
8
Lab Power Supply
SIERRA Alternator / DC/DC Supply
FLIGHT CRITICAL
SIERRA Piccolo Autopilot
16
Ethernet (CAT-5) Passthru
SIERRA Mag. Sensors
15
Isol.
6
Isol.
6
Mag.
3
Power Reg.
5
CDHU (CPU System)
7
Mag Box
4
Laser Altimeter
2
Fluxgate Mag
1
9
Common Payload Data System (CPDS) Assembly
14
Remote User
DGPS/ 900Mhz
Payload Ground Station (PGS)
Repeater for 900Mhz (?)
Internet
12
11
(operates)
Mounted on UAS Exterior
PGS Operator
SIERRA Ground Station
1 Fluxgate Magnetometer (for SIERRA, this component is mounted in the aircraft)
2 Laser Altimeter
3 Geometrics Magnetometer
4 Geometrics Junction Box
5 DC/DC Power Regulator / Conditioner / Distribution
6 RS-232 Optoisolators
7 EAV Lab PC/104 Flight Box
8 Battery (possibly not needed)
9 Common mounting plate (and mounting hole pattern)
10 SIERRA components (offboard)
11 DGPS Antenna (option)
12 900 Mhz Antenna (option)
13 Ground Station (option)
14 Remote Client / Real-Time Internet (option)
15 Geometrics Magnetic Sensor (wing-tip mounted)
16 Flight Critical Systems are isolated from CPDS

Figure 8. CPDS concept architecture

there does not appear to be a requirement to make these connections explicit.

We believe that a goal-oriented approach, specifically evidence-based argumentation, that links the data produced for a COA, to stated COA goals might help solve, to some extent, the problem of the data that is to be included. In particular, by proceeding from defined system-level goals, systematically defining the strategies to achieve those goals and identifying the evidence to be produced (or the evidence that exists), the data to be included is derived as a consequence of safety analysis performed in conjunction with system design. Similarly, we also believe that such an alternative approach would be beneficial, for explicitly conveying how airworthiness goals are achieved.

Towards this end, our methodology can also support system safety assurance through its interface to a process for *safety case* development[f], an alternate method for demonstrating compliance to safety regulations.[1] Furthermore, it explicitly accounts for the heterogenous sources of evidence that have a bearing on safety, and which can be used in the supporting documentation required for a successful COA application.[2]

Acknowledgments

This work has been funded, in part, by NASA contract NNA10DE83C.

References

[1] Davis, K. D., “Unmanned Aircraft Systems Operations in the U.S. National Airspace System,” Interim Operational Approval Guidance 08-01, Mar. 2008.

[f] However, safety case development is not the focus of this paper.

[2] Elston, J., Stachura, M., Argrow, B., Frew, E., and Dixon, C., "Guidelines and Best Practices for FAA Certificate of Authorization Applications for Small Unmanned Aircraft," *Proceedings of the AIAA Infotech@Aerospace Conference*, No. AIAA 2011-1525, 2011.

[3] U.S. Department of Transportation, Federal Aviation Administration, "Airworthiness Certification of Unmanned Aircraft Systems and Optionally Piloted Aircraft," FAA Order 8130.34B, Nov. 2011.

[4] Ippolito, C., Espinosa, P., and Weston, A., "Swift UAS: An Electric UAS research platform for green aviation at NASA Ames Research Center," CAFE Foundation Electric Aircraft Symposium (EAS IV), Apr. 2010.

[5] Ippolito, C., Pisanich, G., and Al-Ali, K., "Component-Based Plug-And-Play Methodologies for Rapid Embedded Technology Development," *Infotech@Aerospace Conference*, No. AIAA-2005-7122, AIAA, 2005.

[6] Scolese, C. J., "NASA Systems Engineering Processes and Requirements," NASA Procedural Requirements NPR 7123.1A, Mar. 2007.

[7] Office of the Chief Engineer, NASA, "NASA Software Engineering Requirements," NASA Procedural Requirements, NPR 7150.2A, Nov. 2009.

[8] Cohen, D., Lindvall, M., and Costa, P., "An Introduction to Agile Methods," *Advances in Computers*, No. 1–66, Elsevier Science, 2004.

[9] U.S. Department of Defense (DoD), "Standard Practice for System Safety," MIL-STD-882D, Feb. 2000.

[10] National Aeronautics and Space Administration (NASA), "Facility System Safety Guidebook," NASA-STD-8719.7, Jan. 1998.

[11] U.S. Department of Transportation, Federal Aviation Administration, *System Safety Handbook*, FAA, Dec. 2000.

[12] Claxton, J. S. and Linton, J. F., "Advanced Range Safety Mission Planning Systems for Unmanned High-energy Vehicles," *11th International Conference on Space Planes and Hypersonic Systems and Technologies*, Oct. 2002.

[13] Leveson, N., *Safeware: System Safety and Computers*, Addison-Wesley, 1995.

[14] Denney, E., Pai, G., and Habli, I., "Towards Measurement of Confidence in Safety Cases," *Proceedings of the 5th International Symposium on Empirical Software Engineering and Measurement*, Sept. 2011, pp. 380–383.

[15] Denney, E., Habli, I., and Pai, G., "Perspectives on Software Safety Case Development for Unmanned Aircraft," *Proceedings of the 42nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2012)*, Boston, MA, June 2012.

[16] Denney, E., Pai, G., and Pohl, J., "Heterogeneous Aviation Safety Cases: Integrating the Formal and the Non-formal," *17th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS)*, Paris, France, Jul. 2012.

[17] Denney, E. and Pai, G., "A Lightweight Methodology for Safety Case Assembly," *Proceedings of the 31st International Conference on Computer Safety, Reliability and Security (SAFECOMP)*, Springer, Sep. 2012.

[18] Williams, A., *Safety Risk Management Document (SRMD) For Establishing a Baseline Hazard Analysis For Operating Unmanned Aircraft Systems (UAS) In Class D Airspace*, Air Traffic Organization, Federal Aviation Administration, Sep. 2008.