MicroPilot®
World Leader in Miniature UAV Autopilots

Requirements-based UAV Design Process Explained

A UAV manufacturer's guide

By Howard Loewen

Unmanned aerial vehicle (UAV) manufacturers depend on design processes, whether they are unstructured and ad-hoc or highly structured and well considered. As the UAVs produced become more sophisticated ad-hoc design processes become insufficient. Presently, more UAV manufacturers than ever have come to realize structured design processes are essential if they intend to develop reliable products.

Furthermore, as regulations develop, manufacturers learn regulating bodies do not tolerate unstructured design methodologies. Moreover, consumers are becoming more sophisticated. UAV end users want evidence verifying their systems are safe. This request cannot be answered by rationalizing away risk. Instead, UAV manufacturers need a robust, structured design process delivering proof that all safety regulations are met.

Unmanned aerial vehicle (UAV) manufacturers rely on a decision making process to incorporate basic sciences, mathematics, and engineering systems into a design plan. Design management uses this formalized design process to convert resources into a final product that meet the needs of the manufacturer and its customers. Without a formal design process, projects can fail and products can nosedive.

A requirements-based design process is the most widely used design process for bringing UAVs into production. This process might not be the most suitable for designing a desk lamp; however, this thorough methodology perfectly addresses the concerns UAV manufacturers face when designing their products.

Requirements describe the functionality of a system while bridging gaps between non-technical design team members (e.g. marketing) and those who implement the design. Clear and accurate communication is essential when designing a highly technical system. Moreover, a requirements-based design system does not tolerate slip-ups. UAVs are airborne products; therefore, manufacturers designing UAVs must use a design process that does not allow a single detail to fall through the cracks. As with any airborne product, even the smallest design error can lead to serious

consequences. A seemingly minor error can contribute to the total loss of a UAV and its payload. In extreme cases, minor errors can even cause injury or even loss of life.

To best convert a marketing concept into a real life system, wish lists dreamed up by high-level management (or high-level requirements) must be broken down into more design-specific instructions. Design teams depend on these decomposed instructions (or low-level requirements) to develop the UAV.

Introduction

One of the greatest challenges UAV design management face is communicating low-level *implementation details* to the team responsible for implementing them. In a requirements-based design process, special care is taken to drive a high-level description of the design through to the *implementation details*. In essence, this high-level requirement is transformed into a list bulleting what is needed on a technical level to carry out the exact system feature. When the methodology is followed correctly, teams from different stages of design, naturally advance each requirement.

Secondly, design teams are responsible for tracking the progress of all aspects of the design process, whether the actions take place during design, implementation, or testing phases. (Design phases are detailed later in this white paper.) With proper requirements-based design software, each action is documented and is retrieved effectively.

Thirdly, design teams need to ensure all critical sections of the design are adequately tested. Since a requirements-based design process relies on simple yet fail-proof checklists, all requirements categorized as critical are flagged *untested* until proper data is submitted.

And lastly, design teams need to view the big-picture of their development process to ensure the design methodology is effective. In the end, when management asks themselves the question, *Does the low-level implementation of this system fulfill the functionality required of the system?*, their answer ought to be *Yes, it does*. A requirements-based design process is synonymous with *quality control* and supports design teams to verify that their efforts match what the UAV must accomplish.

The detailed steps flowing from idea to product in a requirements-based design process are designed to keep teams and individuals on track and in sync. In this way, the process safeguards against strayed efforts and incomplete development of high-level requirements.

UAV manufacturers who rely on a requirements-based design process provide high-quality products to their customers. In addition, these manufacturers enjoy the following benefits:

- Design Management generates highly accurate time-to-market estimates
- Developers experience better communication due to systematic requirements decomposition
- Budget analysts observe lower total life cycle costs
- Requirements and sub-systems/failure modes can be shared among multiple UAVs, simplifying future designs
- Management produces thorough documentation for satisfying regulatory requirements

Additionally, UAV manufacturers have data at their fingertips during the design process detailing the following:

- Which requirements are fulfilled and which are not
- How each requirement is implemented
- How critical requirements are validated

UAV Design Life Cycle

During UAV development, a UAV system moves through a design life cycle. This cycle begins with defining requirements. Next, the system is designed based on these requirements. After design, requirements are validated and then the UAV is pushed through production. The UAV's life cycle concludes when its manufacturer no longer supports the product. A successful design life cycle effectively integrates people, data, processes, and business.

Writing Requirements

A requirement usually consists of a single sentence that describes a feature a system must provide. For example, "The UAV will have an endurance of 2 hours". Writing requirements in this way, accurately communicates low-level *implementation details* to the team responsible for implementing them.

Decomposing Requirements

Requirements are hierarchical in nature with higher-level requirements being decomposed into lower level. For example the requirement above can be decomposed into two sub-requirements:

- The UAV will have a two-liter fuel tank.
- The UAV will use an engine with a cruise fuel consumption of no more than one liter per hour.

Requirements are decomposed until they reach a level of detail those responsible for implementing the project can use to proceed. Again, these steps support clear communication between design management and design teams.

Implementing Requirements

Once all of the high-level requirements are approved and decomposed into low-level requirements, design teams then implement (or satisfy) each requirement. In some cases, requirements are satisfied in multiple ways. Safety requirements, for example, might need appropriate settings within the UAV's autopilot, an additional section in the maintenance manual, and operator training.

Validating Requirements

Once requirements are implemented, design teams validate (or test) each requirement. Since a requirements-based design process revolves around the systematic breakdown of requirements, the process itself generates a comprehensive list of items to be tested. After implementation, every requirement presents a method of satisfaction, whether it is an autopilot setting, manual section, piece of equipment, or a combination of methods. These implementations are then tested against the original requirement to ensure that the implementation works properly.

During validation, UAV manufacturers test both high-level and low-level requirements depending on type. For example the high-level requirement, "The UAV will have an endurance of 2 hours", can be validated by demonstrating that the UAV can maintain flight for this period of time. Low-level requirement falling beneath this high-level requirement could also be tested. However, unless more requirements exist that depend upon the low-level requirements, validating only the high-level requirement is more efficient.

In some cases, high-level requirements simply cannot be tested. They might be too vague, (i.e. “The UAV will handle all likely failure conditions”), no method to prove its validity exists (i.e. “The UAV will have a lifetime of 1000 hours”), or testing is too expensive. Flight-testing teams will instead validate each of the low-level requirements that fall beneath these high-level requirements. In this way, flight-testing teams do their diligence to satisfy these high-level requirements.

During testing, design teams may discover flaws in their original requirements. For example, when validating the requirement “The UAV will operate in a wind speed of 25 kmph”. If the original cruise speed requirement was for a cruise speed of 30 kmph, then the process of validating the UAV’s ability to operate in a 25 kmph wind would certainly identify a serious problem with the UAVs ability to fly upwind. As a result, the design team might have to add the following requirements:

- The UAV will automatically increase cruise speed by up to 10 kmph when flying into wind.
- The UAV will automatically measure wind speed and display the speed on the ground station.
- The ground station will warn the operator if the wind speed exceeds 20 kmph.
- The ground station will automatically estimate the time to return, taking into account the wind and display this information to the operator.
- The ground station will automatically warn the operator if the fuel/battery is not sufficient to return.

A robust requirements management software solution allows users to satisfy requirement in more than one way.

As part of quality control, design teams trace each *implementation detail* back to its originating requirement. Occasionally, a method of satisfaction (such as an autopilot option, source code, or manual section), will not link back to a high-level requirement. In these cases, design teams evaluate whether the implementation is essential to the functionality of the product. If the *implementation detail* is necessary, a new requirement is added to justify the satisfaction method. If the *implementation detail* is not required, then it is removed. Not only does this crosschecking ensure the final system is solid, it also minimizes the number items that need to be tested.

Ideally, implementation is complete when testing begins; otherwise tests performed for altered requirements could be deemed immaterial and need to be redone. Validating requirements is one more way a requirements-based design process safeguards that low-level implementation of the UAV system fulfills the functionality required of the system.

Traceability

A manufacturer's ability to trace their steps back to mistakes made is its best means of accountability. Generally speaking, people are more careful when they are accountable for their actions. When driving a UAV from concept to production, a requirements-based design process is particularly helpful for tracking all actions during the design process and providing reports on how requirements have been fulfilled and tested. Robust requirements-based design process software documents every change made to a requirement, providing traceability. In other words, those reviewing its history files know exactly who originated, modified, implemented and tested each portion of the design.

Making the Commitment

In order for a requirements based design process to succeed everyone on the design, implementation and test teams must be committed to the process. There is no sense in writing high level requirements if design management doesn't get around to decomposing them into low level requirements. There is no use in having low-level requirements if the implementation team doesn't look at them when they implement the design. If the test team doesn't bother to use the requirements when developing their test plan, or link test data to requirements then, again, the system breaks down. If management says use requirements based design but then does not allow members of the design, implementation and test teams the time they need to decompose, implement and test requirements the system will be very quickly abandoned.

A successful implementation of a requirements-based design methodology can look relatively effortless. In fact, this process calls for a tremendous amount of discipline. UAV manufacturers who consider implementing a requirements-based design process need to evaluate whether they have the correct resources. Firstly, this design methodology adds overhead to the development and implementation process. Managing requirements imposes additional paperwork and/or data entry. A requirements-driven solution might not be cost effective or necessary for a UAV company who intends to create a very basic system.

Additionally, commitment and focus are essential to successfully carrying out a requirements-based design process. UAV manufacturers need to gain a commitment from across their organization — management, the design team, and the test team. UAV manufacturers need to decide whether or not their organization is sufficiently disciplined in this regard.

Furthermore, all individuals involved must demonstrate focus. For example, if management lacks the ability to visualize exactly what their customer's need, then they might write high-level requirements for an undesired system. Due to the orderly development of requirements going forward, many sub-requirements will be written for this undesired system. When the error is finally recovered, much time and money is lost. Moreover, requirements are typically written in English or other spoken language. As such, requirements can be subject to interpretation, misinterpretation, or be incorrect due to misuse of the language.

Conclusion

The way in which a manufacturer designs a UAV ultimately determines the system's reliability. If failures occur during the design process costly parts are often damaged and on rare and unfortunate occasions, people are injured. Careful attention needs to be placed on requirements development, system design, and thorough and complete testing.

Investing in a standard and efficient requirements-driven design process helps UAV manufacturers bring their product to market swiftly, ensures the development of high-quality systems, and generates more accurate and timely quotes for their customers. Moreover, this methodology curbs design costs by allowing teams to share requirements and sub-systems/failure modes among multiple UAVs.

About MicroPilot

With 750 clients in 65 countries, MicroPilot is the world leader in miniature autopilots for UAVs and MAVs. MicroPilot is the first UAV autopilot manufacturer to bring to market a sub 30 gram autopilot, triple redundant autopilot, and full-function general purpose autopilot. MicroPilot offers a family of lightweight UAV autopilots that can fly fixed-wing, transitional, helicopter, and Multi Rotor UAVs. MicroPilot also provides complementary products such as the XTENDER^{mp} SDK, trueHWIL, payloads, and catapults.