THE APPLICATION OF CONTINUOUS INTEGRATION AND TEST METHODOLOGIES TO ROTORCRAFT HUMS

Paul Dunning, Paul.Dunning@ge.com, GE Aviation Digital, Southampton (UK)

Mateusz Nowik, Mateusz1.Nowik@ge.com, GE Aviation Engineering Design Centre, Warsaw (PL)

Lukasz Nyczkowski, Lukasz.Nyczkowski@ge.com, Łukasiewicz Research Network – Institute of Aviation, Warsaw (PL)

Christopher Mead Chris.Mead@ge.com, GE Aviation Digital, Southampton (UK)

Abstract

Continuous Integration and Test (CIT) is a mature technique used to support Agile software development practices and it has seen increasing adoption for avionics software development over the last 10 – 15 years. However, point solutions are usually adopted for avionics CIT as standard frameworks for implementing it do not exist. Test PASS, developed by The Warsaw Institute of Aviation (WIA) in collaboration with General Electric Company Polska (GECP) is proposed as a framework for CIT in an embedded, safety critical environment. Its application to GE's next generation HUMS is described as a challenging use case that fully demonstrates the framework's flexibility and capabilities. The challenges of implementing a CIT approach for HUMS are presented as are the solutions enabled by Test PASS. The benefits to the GE's next generation HUMS product including improved robustness and development effort efficiencies are described.

1. INTRODUCTION

It is arguably the requirement for increased safety that has resulted in the vast majority of HUMS installations to date, where civil operators are increasingly required to meet related regulatory requirements and military operators are acknowledging a duty of care. The system's safety benefits were demonstrated by a number of studies, as recorded by the UK Civil Aviation Authority (CAA) [1], [2] and this led to a requirement for its installation for helicopters operating in the North Sea carrying over 9 passengers [3]. By the end of the 1990s, the UK Civil Aviation Authority (CAA) claimed that "HUMS…has already reduced fatal accident statistics".

The implementation of HUMS was also driven by the motivation to save costs through maintenance and operational benefits. Many such benefits are now established and well documented [4], [5] and HUMS is now more widely seen as a worthy investment to new and aging, civil and military platforms. Focus has been applied to unlocking the economic benefits from HUMS generated data and it has been shown that multimillion dollar yearly savings can be achieved with the judicious application of HUMS [6]. Therefore, to ensure these continued safety and cost benefits, it is important that the system is robust and unexpected costs arising from development errors are minimised.

Copyright Statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ERF proceedings or as individual offprints from the proceedings and for inclusion in a freely accessible web-based repository.

Finding errors early means that correcting them is less costly and the resulting product is more robust. In a typical waterfall development lifecycle the cost of fixing errors at the test/integration phase has been shown to be up to four times higher than during the development phase [7]. Continuous Integration and Test (CIT) is a development practice where integration occurs daily. Builds are automated, and integration issues are discovered as quickly as possible. As a key enabler for Agile software development, CIT methodologies have been widely adopted and have transformed how software is integrated by "shifting left" testing so that it is concurrent with development.

CIT is now a mature technique within software engineering and off-the-shelf tools and processes exist to implement it within that industry. However, a challenge for safety critical avionics, and HUMS in particular, is the significant amount of hardware in the loop, the practicalities of which hamper the application of these readily available tools. As such, diligent engineers eager to implement best practice for the benefit of their current project find point solutions. While these are effective, the benefits on one project are not often easily translated to the next, and another point solution is created.

The Test PASS research project explores how to test more efficiently and more automatically in an avionics environment, in order to provide a common test framework to realise the benefits of CIT for avionics. The Warsaw Institute of Aviation (WIA) in consortium with General Electric Company Polska (GECP) is developing the Test PASS framework. In collaboration with GE Aviation's multi-national team they are using it to apply CIT methodologies to HUMS development to reduce development schedules and improve robustness. *Quid pro quo*, the application to HUMS benefits Test PASS by ensuring that the features that it offers are easily translatable between products. As Test PASS aims to avoid the need for point solutions, its application to HUMS as a system with a unique set of needs is seen as a good test case.

Presented at 45th European Rotorcraft Forum, Warsaw, Poland, 17-20 September, 2019

This work is licensed under the Creative Commons Attribution International License (CC BY). Copyright © 2019 by author(s).

1.1. Goals

The research effort has four goals:

1. To leverage CIT processes and methodologies for the benefit of HUMS development.
2. Apply the Test PASS framework as a means to achieve the new processes defined under item 1.
3. To demonstrate the capabilities of Test PASS as a generic framework for automated testing of avionics
4. To build upon the already strong international collaboration partnership and develop HUMS application knowledge within GE Aviation/WIA Engineering Design Centre (EDC) in Poland.

2. HUMS SYSTEM OVERVIEW

Since GE introduced the first civil certified HUMS in 1991 it has matured into an important system for rotorcraft. HUMS is now normally specified as standard fit on medium and large types for both civil and military helicopters either in response to legislation or in acknowledgement of the duty of care for personnel. HUMS provides safety benefits by giving a deeper insight into the aircraft's operations and the health of key components. On a helicopter one of the most important components monitored is the drivetrain and rotor system as these provide a non-redundant, single load path for maintaining continued flight. A failure in this system can be catastrophic but by understanding its dyamic behaviour expressed in its vibration signature, component health can be monitored and incipient faults detected. The rotor and drivetrain are a significant source of vibration which is used by HUMS to monitor the health of these subsystems. A virtual net of accelerometers is placed over the helicopter to capture this vibration which is then analysed by the HUMS algorithms to produce Condition Indicators (CIs) which, trended over time, can be used to determine faults in gears and bearings and imbalance in transmission shafts.

In addition to providing drivetrain health monitoring and Rotor Track and Balance (RTB), HUMS monitors the long term usage of the airframe and detects operational exceedances against airframer defined limits. It does this using sensor data and flight/engine data feeds from digital buses. As a result HUMS has sometimes been co-located with the Cockpit Voice and Flight Data Recorder (CVFDR) to take advantage of the overlap between the systems' input capabilities.

2.1. GE's HUMS

GE Aviation's latest generation HUMS represents a step change in capability while still offering Form, Fit and Function (FFF) compatibility with previous installations. As with previous HUM systems, it is capable of being physically co-located in the same Line Replaceable Unit (LRU) as the CVFDR or operating as a standalone system.

In its default configuration the system provides 24 accelerometer inputs and 8 tachometer inputs for RTB and powertrain health monitoring with vibration being sampled at >200kHz. Utilizing modern computing architectures means that vibration inputs can now be continuously and simultaneously sampled which greatly reduces the time taken to perform a full transmission scan. A full transmission scan could previously take 45 minutes and this led to dedicated "HUMS flights" for helicopters which were typically flown on short operations. Scans with GE's latest generation HUMS are typically limited by the speed of the slowest rotating component (not the capability of the HUMS itself) meaning that in 5 minutes or less a full transmission scan can be completed. This eliminates the need to acquire

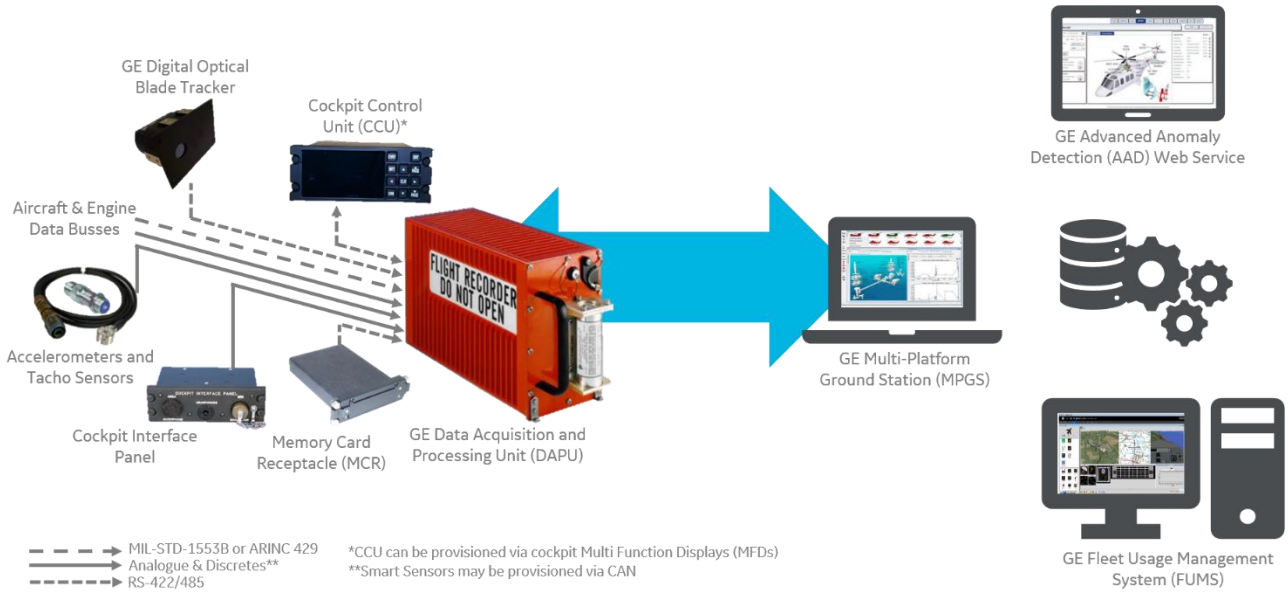

Figure 1: GE's combined HUMS CVFDR

Presented at 45th European Rotorcraft Forum, Warsaw, Poland, 17-20 September, 2019
This work is licensed under the Creative Commons Attribution International License (CC BY). Copyright © 2019 by author(s).

specific drivetrain data due to sparse data coverage and thus greatly reduces the need for HUMS flights. It also means that multiple scans can be performed on a single flight giving multiple data points per operation improving the monitoring of the transmission and providing enhanced safety benefits.

Regime analysis, usage monitoring and exceedance detection is provided by ARINC 429 and MIL-STD-1553 connections which can be further augmented by over 100 Analogue/Discrete inputs or outputs. These digital and analogue inputs can be recorded across the full flight. Whereas before an operator may be limited to seeing a snapshot of data around an exceedance for a limited set of parameters, the full flight's worth of data is available for analysis. Exceedances themselves can be operator defined via the system's configuration tools.

Controller Area Network (CAN) I/O is provisioned for connection to smart vibration sensors which provide the opportunity to reduce overall system weight by eliminating wiring. Peripheral equipment can also be connected via Ethernet, RS-485 or RS-422 a key piece of which is GE's digital Optical Blade Tracker (OBT) which is used to determine difference in blade dynamics for the RTB function. The Cockpit Control Unit (CCU), sometimes integrated with other cockpit displays, is another critical piece of peripheral equipment that provides for control of the system by the crew.

Ultimately, HUMS data is collated and transferred to the GE's Multi-Platform Ground Station (MPGS) which provides HUMS data analysis to support day to day operations. Transfer of data to and retrieval of configurations form MPGS is achieved via data cards or the provision of wireless capability. The MPGS not only allows the review of gathered HUMS data, it is designed to compliment the day to day helicopter maintenance activities of operators, seamlessly integrating with operational policy.

Hence, it can be seen that HUMS presents a significant challenge for Integration, Verification and Validation (IV&V) as it is a highly connected, highly configurable system with a wide breadth of functionality. Key areas of this functionality, such as transmission health monitoring and RTB also require specialised V&V methods to ensure that they are operating as expected. GE's next generation HUMS system has further compounded this by offering greater depth of capability and more advanced features, such as the concurrent acquisition of vibration data. And so, just as the system continues to develop so must the techniques used for its development and IV&V.

2.2. HUMS IV&V

As is more generally true for avionics, traditional HUMS development is typified by "Waterfall" or "Vee" lifecycle development leading to a "Big Bang" integration. In this environment non-obvious design errors and emergent behavior are not uncovered until rectification is significantly more expensive [7].

The INCOSE Systems Engineering Handbook [8] describes eight possible approaches to integration and notes that all are valid approaches and they may be blended to suit the system under development. Of the integration methods listed therein, GE's traditional HUMS integration practices could best be described as a combination of *Global Integration* (aka Big Bang) where integration occurs when all the delivered system elements are available and *Integration "with the stream"* where the elements are integrated as they become available. Of the former method the INCOSE Handbook notes that there can be difficulty in detecting and localizing faults and of the latter it states that "*[Integration "with the stream"] should be reserved for well-known and controlled systems without technological risks*".

The first generation of GE HUMS were implemented on dedicated circuit cards and integrated with other on-aircraft systems such as a Digital Flight Data Unit (DFDU). As such they were tested at the circuit card level, then at the system level and finally at the aircraft level, progressing up the right-hand side of the Vee. As HUMS was a novel system, GE was involved at every step and so gained a good understanding of how circuit card level behavior manifests at the aircraft level. Verification of the system was largely completed during flight tests, checking the operation of each function against its specification and simultaneously validating the system was performing as intended. The system was designed from the outset to be easily re-configured in the field so that, for example, RTB co-efficients could be fine-tuned to ensure the best response during aircraft tests. The on-aircraft validation also included validation of transmission Condition Indicators (CIs), collection of raw vibration data, and full excitation of flight data parameters across their valid range. On completion of these tests the system was certified.

The implementation of GenHUMS for the UK MoD allowed this on-aircraft experience to be exploited and further developed. MoD practices now mandate that their regulations are "*as civil as possible, as military as necessary*" [9]. However, during the development of GenHUMS, by the military regulations in force at the time the HUMS configuration was considered "data" and so *ad hoc* on aircraft validation was possible without full, for credit verification. Ultimately this resulted in fewer errors in the fielded configuration as emergent behavior had been detected and corrected during the development phase. Conversely, with more stringent safety regulation now in place the opportunity for early on-aircraft validation is greatly reduced and a certificate of design must be issued before flight test. This observation is not unique to the UK MoD, rather it reflects their alignment to civil certification practices. The effort to complete an engineering release is as onerous as a full production release and hence, only production releases with full lifecycle testing are released to the UK MoD. On aircraft testing still occurs and errors are caught at this stage, but it occurs after GE has completed its release and the rectification of errors is more costly.

HUMS configurations undergo both static inspection and dynamic testing. Specific tests are written for each and a small team of engineers with specialist knowledge completes the task. This limits the opportunity for test independence[1] and creates resourcing challenges.

[1] Not required at DAL D but still good practice.

Presented at 45th European Rotorcraft Forum, Warsaw, Poland, 17-20 September, 2019
This work is licensed under the Creative Commons Attribution International License (CC BY). Copyright © 2019 by author(s).

Static inspection involves review of around 50,000 separate data items which is a time-consuming task and, in lieu of a qualified tool, must be completed manually. Dynamic testing involves rig based simulations which ensure the system completes all its measurements and flight data acquisitions, but due to the complexity of running the tests excitation of each accelerometer channel with representative waveforms, with a synchronized tachometer input is completed only when necessary. An automated system would make this process far more routine.

The development of the next generation system has experienced two further effects have compounded IV&V challenges. Firstly, as an FFF replacement which is directly compatible with the previous system, it is expected to work first-time when installed on the aircraft. While flight tests will occur, they are expected to be more of a formality and the opportunity for a campaign of on aircraft tests to validate system performance is eliminated.

Secondly, customers are now far more familiar with the operation of HUMS. GE still supports V&V at every level, but it is now far more typical to have a relatively stable requirements specification and Interface Control Documents (ICDs). This document based approach, while encouraged by aerospace standards delays the detection of errors.

2.3. Summary of challenges for HUMS IV&V

We have demonstrated in section 2.1 that HUMS is a complex system and in section 2.2 that constraints on IV&V for the system create practical challenges that must be solved. Table 2-1 summarises these findings.

Table 2-1: Challenges for HUMS IV&V

Id	Challenge	Effect
1.	HUMS IV&V is a time consuming, manual task.	The addition of further test cases must be carefully considered.
2.	Traditional HUMS integration methods can make fault localisation difficult	Testing at lower levels would be beneficial, but increases test effort.
3.	An expectation of improved ROI creates development cost pressures.	Detecting errors earlier makes them cheaper to resolve, but a strategy is required to achieve this.
4.	HUMS IV&V is a specialist task requiring specialist resources	Opportunities for distributing work is limited. Test automation is non-trivial.
5.	As a FFF replacement there is an expectation that the system works first-time.	Robustness and performance must be understood and verified during development.
6.	Opportunities for on aircraft validation are reduced.	The scope of testing at the software/ system level must be increased.

As a result, we have a situation where more testing at a lower level would be beneficial to system performance and robustness and is essential in some cases, but the addition of further testing is difficult due to the specialist capabilities required and the perpetual challenge of controlling costs. Simply doing more testing in the same way that it has always been done is not an option. A change in approach is required.

3. WHAT IS CONTINUOUS INTEGRATION AND TEST (CIT)?

The CIT approach is best understood from the point of view of the lifecycle it supports and the issues that it is attempting to resolve. Traditional development practices such as Waterfall or Vee lifecycles emphasize upfront effort to define the system and leave test activities until the end. Note that there is a distinction between test and verification. Waterfall and Vee lifecycles do not prevent V&V on design artifacts and time spent verifying the design in the system definition phase is rarely wasted. However, there is no substitute for real world experience and, as described by Larman [10], Waterfall[2] has the following drawbacks for integration and test:

- It delays higher risk and difficult tasks,
- It is poorly suited to deal with changing requirements,
- Upfront schedules and estimates are unreliable,
- Late integration is encouraged.

The Agile software development methodology was developed as a reaction to “fixed process” mindsets of which Vee and Waterfall lifecycles are arguably examples of. CIT supports at least two of the key principles of Agile [11]:

- “*Deliver working software frequently”*
- “*Working software is the primary measure of progress”*.

In practice this means creating working prototypes early and often. It also means integrating software components frequently during development rather than when the full development is considered complete. Hence, learning is accomplished early, technical risk is reduced, and difficult tasks are accomplished in a timely manner. Changing requirements can be adopted as the timescales between development steps is greatly reduced from many months between releases to a couple of weeks between iterations. Upfront schedules and estimates are not necessarily more reliable, but

[2] While distinctly defined, for the purposes of this discussion the Waterfall and Vee lifecycles are equivalent as they are both staged and both place test after development.

Presented at 45th European Rotorcraft Forum, Warsaw, Poland, 17-20 September, 2019
This work is licensed under the Creative Commons Attribution International License (CC BY). Copyright © 2019 by author(s).

unambiguous feedback about their reliability is available more quickly so that they may be adjusted. Finally, regular, almost daily, integration with the mainline code repository is essential so that incompatibilities between software units are uncovered while they can be easily fixed.

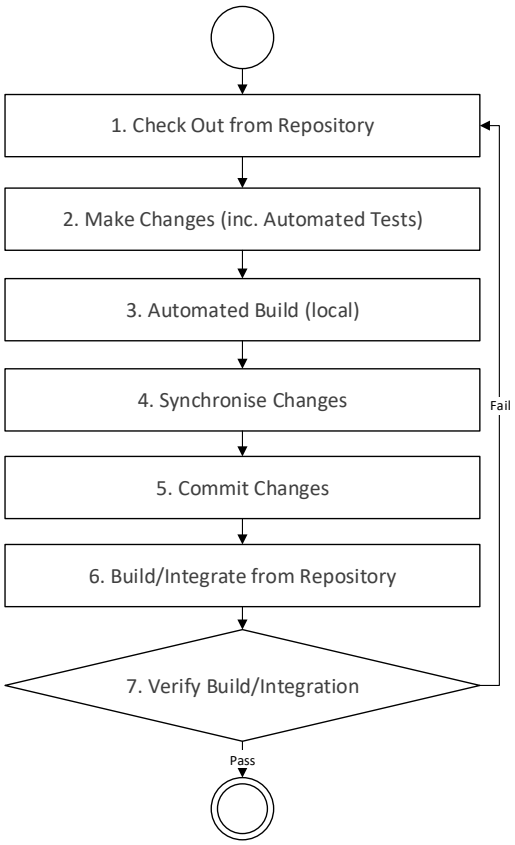

Figure 2: CIT development process [12]

Fowler [12] describes the process of developing a feature in a CIT environment which is summarized in Figure 2. Code is checked out from a central repository, changes are made and the software is built on the developers local machine. Then, as others have usually been making concurrent changes, the changes are synchronized with the latest repository version, essentially repeating steps 1 to 3 until no further errors are found locally. Changes are then committed to the repository. This process will be familiar to any software developer, whether they adopt CIT or not. They key differences are in the scope of changes (step 2), the changes that are made and what happens to the changes when they are committed back to the code repository (steps 6 and 7).

Whereas in a Waterfall/Vee development a complete software unit may developed from scratch before integrating with other units, the changes made before integration under CIT are typically very small. The changes are also made not just to the code but to the automated tests that support the code. This is a key point. CIT requires that a large proportion of tests are automated into the software and that these tests are maintained along with the code base. Finally, once the changes have been committed back to the repository an automated build and test is initiated either on check in or overnight (when changes to the repository are typically less frequent). Critically, the task of making a change is not complete until the automated build and test completes without failures.

Figure 3 compares the level of test effort throughout a project lifecycle for a pure Waterfall/Vee, an iterative Waterfall/Vee development with multiple releases (as is more typical), and a CIT based development. For Waterfall/Vee, large test programs are accomplished once development completes and before every release as was described in section 2.2. Of course, project pressures normally mean that the time available to complete this campaign of testing is reduced. Conversely with CIT, as testing occurs throughout the project the amount of testing prior to a release is reduced. This does not mean, however, that the total amount of testing during the project lifecycle is cut. Quite the opposite; CIT requires a great deal more testing and this is another driver towards test automation.

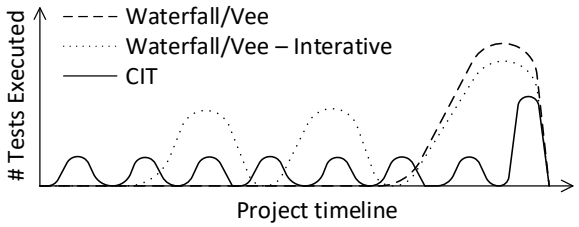

Figure 3: Comparison of amount of testing during project lifecycle

CIT is not a new concept for avionics software development and as a subject intertwined with Agile software development, interest in CIT has grown concurrently with interest in Agile practices. In a 2009 survey of Agile techniques and adoption in the aerospace domain VanderLeest and Buter [13] noted "*We have been able to help implement and work with continuous integration systems on multiple DO-178B Level A programs*" and that "*The use of continuous integration systems has allowed our teams to immediately identify any issues where a change to one component impacted another component"*.

Although it is not novel, there are challenges in applying CIT for avionics software development which is characterized by the use of embedded software. Emulated hardware environments are used for development and development testing, but integration with the target hardware is a still a risk. Extending CIT principles to the target hardware to reduce this risk requires that common challenges such as automating the loading and control of software, the application of power and physical reset of the hardware are solved. Although these challenges are not unique, the solutions to them often are.

In the authors' experience automated testing and adoption of CIT for avionics equipment is becoming common, although not yet commonplace. However,

Presented at 45th European Rotorcraft Forum, Warsaw, Poland, 17-20 September, 2019
This work is licensed under the Creative Commons Attribution International License (CC BY). Copyright © 2019 by author(s).

point solutions for each project are often adopted and therefore, an objective for the application of CIT to HUMS is to minimize point solutions and maximize the use of standard tools to gain maximum re-use benefits, not just for HUMS but for other GE Aviation avionics developments.

3.1. Enabling CIT for HUMS

To establish a CIT environment for HUMS it was necessary to establish how software could be developed on representative target hardware as early as possible, and identify which tests represented a clear opportunity for automation.

Currently over 70% of GE's HUMS software high level test cases are automated or planned for automation. Even when accounting for test infrastructure and implementation costs on a single project this realizes test cost efficiencies and allows the scope of testing to be vastly increased. Using the established infrastructure future projects can expand this further. It is worthy of note that automated tests were not originally planned for certification credit and that the inclusion of for-credit tests was not necessary to realize an ROI for HUMS automated testing.

Whilst cost efficiencies were desirable, schedule reductions were of greater value to the project. Removing constraints on human rig time (e.g. simultaneous/overnight running) saved time and will also improve quality as multiple test runs will increase the robustness of the product. In effect we are defeating the "iron triangle" of project management, by offering reduced schedule and cost, and increased quality for a fixed scope and cost.

The unique challanges for automated testing for HUMS are to be found in the complex nature of the product and its interfacing requirements. HUMS acquires very high frequency (>200kHz) vibration signals. Recall that a key feature of next generation system is the ability to simultaneously acquire accelerometer signals to reduce transmission scan times. This required the careful integration of high-fidelity signal generator controls which are used to replay drivetrain defect signals and verify GE's proprietary diagnostics.

Test PASS was the tool selected to overcome these challenges and enable a CIT based development for HUMS, employing the architecture shown in Figure 4.

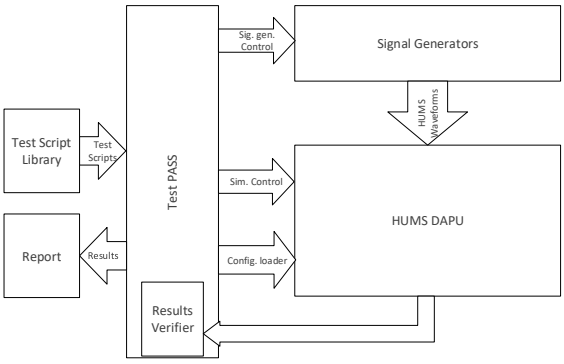

Figure 4: HUMS Automated Test Architecture

4. TEST PASS DESCRIPTION

4.1. Test PASS – research motivation

4.1.1. Problem Statement

Large engineering companies have a huge potential which derives from experience and diversity of their workers, funds allowing for use of specialized lab equipment and the latest software, and huge amount of data collected over many years to learn from. And yet, there are inefficiencies that their scale brings and must be dealt with. One of the most commonly encountered is internal communication or more precisely the sharing of knowledge.

Even though sharing is actively encouraged through the introduction of collaboration platforms these tool very often cause information noise and don't necessarily help in finding desired information. Indeed, a recent Harvard Business Review article states that even in this modern, collaboration enabled environment 55% of employees turn to their colleagues when learning something new [14]. This leads to a situation where, in need a solution to their problem, an engineer would eventually give up searching for advice and they will try to solve the problem by themselves or with their small team.

Let's assume that the challenge is to select a tool suite for the next program. Experienced engineers will prefer tools they already know and trust rather than learn to use a new one. Young engineers often opt for the latest, lightweight, open-source programs. No two teams are composed the same and this diversity leads to a situation in which isolated groups of engineers are constantly dealing with almost the same problems but coming up with a slightly different solution.

This creates several derived problems. Initially money and time are waisted "reinventing a wheel". Multiple, project specific solutions must be managed, and differences between development/testing environments cause long learning curves for people changing projects. Finally, company may end up with a variety of different software licenses, that are rarely used.

4.1.2. Proposed Solution

Considering characteristics of avionics projects and the fact, that they are usually accomplished by large companies, a team consisting of WIA and GECP engineers applied for grant to *National Centre of R&D* in Poland to research a topic of efficient testing in safety-critical environment. As a result, funding for research was granted in 2017 from EU program called "*Operational Programme Smart Growth*". The goal of the project is to identify opportunities to improve IV&V practices and propose solutions that would make it more efficient. The expected outcome of the project is a know-how and software product, that can be used to provide a solution, consisting of both services and tools, diminishing cost of testing to the degree that it is feasible to do it early, automatically and continuously. This maps well to the objectives of CIT and so the research effort reported in this paper contributes to the higher level programme.

Presented at 45th European Rotorcraft Forum, Warsaw, Poland, 17-20 September, 2019
This work is licensed under the Creative Commons Attribution International License (CC BY). Copyright © 2019 by author(s).

External funding allowed the Polish team to spend considerable amount of time objectively looking at different projects' needs and finding common causes of problems. This is relatively unusual as research budgets are often allocated to less prosaic pursuits and normally engineering activities are funded project by project and with a very little space for process/tools modification especially with perennial budget and schedule pressures.

Observing how avionics projects are executed led the engineers to following recommendations to change the way testing is done:

- As far as possible, the solution should make use of existing tools whether they are off-the-shelf, bespoke or homegrown,
- There must be a group of people, that understands how to meet projects' needs using these tools,
- There must be a tool that provides a unified interface for testers, so they can focus on writing tests and not developing a deep understanding of the technical aspects of the tool chain.

The proposed solution has two elements, a test tool supported by a service. The service leads to clear division of tasks between people and their roles:

- Project ***Technical Leads***, typically experts in the system to be tested, are responsible for determining what to test, using which method and what data to collect from testing process. They should work with Integrator to confirm structure of proposed test environment.
- ***Integrators*** are responsible for knowing what tools are available and being familiar with their characteristics. They propose the means to test a product's functionality and obtain the test data required by the technical leads. They should work with project technical leads to propose structure of test environment and bring it to operational state.
- ***Testers*** understand product's requirements, the test environment interfaces and write/conduct functional tests based on it.

This concept is captured in Figure 5.

The second part of the solution is the tool that consolidates all other tools under one umbrella and provides an test interface specifically tailored to the testers' needs. This tool is called Test PASS, which stands for "***Test Platform for Safety-critical Systems***".

The platform is built as a set of different components communicating together over standard protocol that is agnostic to physical interfaces. This architecture reflects the Unix philosophy which favours composability over monolithic design [15]. The platform isn't just one application but rather several scripts/programs, that can be put together in a client's environment in the way it satisfies their needs. Certain elements can control an embedded system, PC or entire laboratory rack system – all being visible as the nodes of the same generic type.

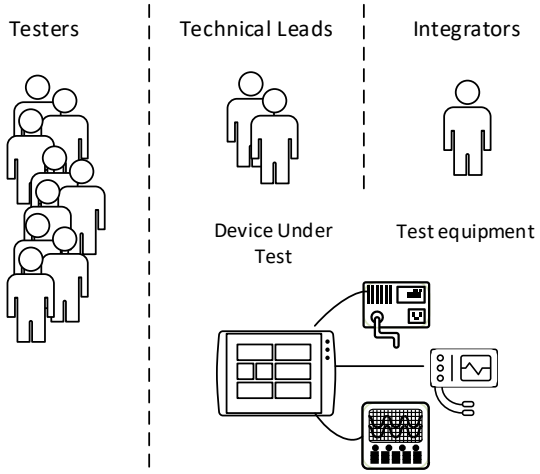

Figure 5: General division of responsibility implied by Test PASS framework. Testers are isolated from test environment configuration details, Technical Leads are the focused on what to test should be tested in DUT, Integrators organize test equipment and run set up test environment.

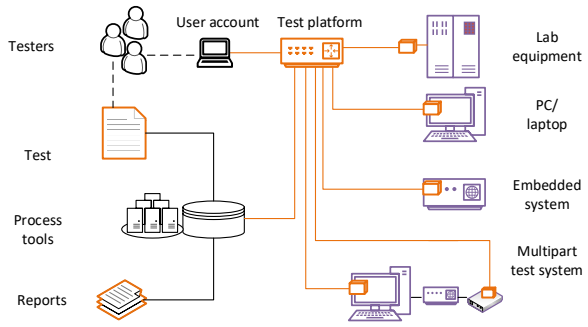

Figure 6: Test platform (orange) operating on different targets (purple), interfacing existing process tools (black).

By default the platform provides a Command Line Interface (CLI) which is suitable for automation in almost every case. It provides for simple of chaining outputs of one program to inputs to another, scripting, freedom of modification and adding/exchanging components. A CLI is the quickest way of achieving the goal of running all the tests with one click, especially for a complex test environment, where many different technologies are used. However console programs have their drawbacks too – it's not easy for an inexperienced user to set them all up and invoke commands in the correct order. Some effort is needed to become familiar with the environment to get started and for that reason Test PASS is meant to be set up at client's side by trained personnel. This is an example of the implementation of the service concept and prevents anyone burdening anyone with unnecessary effort. Once this is done, setting a tester's workspace is trivial activity.

4.2. Test PASS functionality

As discussed above, Test PASS's main task is to provide a consistent way of executing automatic tests and abstracting technical aspects of test environment from the tester. To achieve this the tester is being provided

Presented at 45th European Rotorcraft Forum, Warsaw, Poland, 17-20 September, 2019
This work is licensed under the Creative Commons Attribution International License (CC BY). Copyright © 2019 by author(s).

with the following test Application Programming Interface (API), split into groups as listed in Table 4-1.

Table 4-1: Test PASS API Groups

API Group	Functions
Basic API – related to test results analysis	• Transfers the results of checks, traceability markers, and identification of specific test procedure steps to a results report,
Extended API – Platform dependent	• Provides a mechanism for synchronization, meta-data transfer, reading input files, writing output files, etc.
Functional API	• Allows control lab of equipment (e.g. signal generators, power sources, robotic arms), • Supports communication interfaces external to the tested system, • Supports industry standard communication protocols (ARINC664, RS422, etc.) • Supports various data formats to ease analysis and verification (XML, JSON, etc.)
Client/project Specific API – functions that are specific to client test environment.	• Product specific information exchange protocols • Data manipulation or verification functions • Automatic configuration of Unit Under Test • Wrappers for functionality of a tool/test system that is used as a component

With the functionality listed in Table 4-1, Test PASS provides a means to run a series of tests in different modes and configurations. The platform can execute a multipart test, synchronizing execution of every part involved using a logical connection between the Unit Under Test (UUT). This may involve, for example, connection to a product specific protocol on a communication interface that is connected to UUT with PC computer, while driving inputs of UUT using lab equipment. An example of this is shown in Figure 7 where automated HMI control together with image verification is used to verify instrument displays. A more complete description of this application of Test PASS is provided by Stanislawski [16].

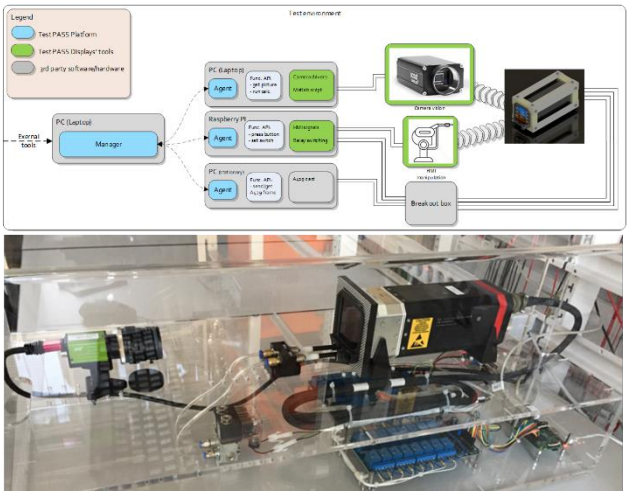

Figure 7: Example of complex test enviroment realized with "multi-part" functionality provided by Test PASS. Three independent subsystems are realizing communication, HMI manipulation and Image verification eliminating need of any human intervention during test.

Test PASS also provides a coverage collection mechanism that uses 3rd party instrumentation tools and monitors. It allows the integrator to prepare environment (application images, coverage collection tools configuration, etc.) so it's common for all testers working on the project and there's no dependency between this configuration and tester's workspace.

The tester can choose to run their set of tests in order to get percentage of coverage measured for their module. Moreover, 3rd party elements can be exchanged by system integrator causing no changes to tester's everyday routine. This independency between the way the test is written, and which tool is used to gather metrics enhances portability of tests between different project. This enables a true Component Driven Development of products in the organization. Test PASS can also automatically run a test addressed to a certain component that is available on many different targets ("multirun mode").

4.3. Test PASS Support for Avionics CIT

CIT is impossible to implement without test automation. As described in Table 2-1 HUMS and avionics related products frequently need sophisticated lab equipment to confirm that system behaves as expected. This creates a need for CIT tools to be able to automate complicated scenarios with multiple steps and checkpoints often involving the operation of lab equipment. As an example one of test scripts for HUMS has 117 separate check points. This can be problematic for automation and creates temptation to leave some tests as manual checks. However, yielding to that temptation will result in those manual tests being executed less frequently and an interruption in the flow of automation. This would eliminate many of the time saving and robustness benefits of test automation.

Test PASS addresses this problem via the functional and project specific API groups (Table 4-1). Together with its

Presented at 45th European Rotorcraft Forum, Warsaw, Poland, 17-20 September, 2019
This work is licensed under the Creative Commons Attribution International License (CC BY). Copyright © 2019 by author(s).

built-in mechanisms for multi-part test operation the Integrator can easily build test environment which overcomes any automation problems including system communication, lab equipment operation, image recognition, HMI automation and so on.

Another common issue is the availability of hardware resources. Avionics equipment is expensive and this limits availability of early prototypes typically to a few units, meaning they must be shared between developers. Test PASS provides mechanisms for sharing these resources ensuring that build-up and tear-down actions are executed when needed. Remote access and version control interfaces assure that appropriate tests are executed, and results are stored in dedicated artefacts management tool. Tester can also use the option of test coverage collection for the module under test.

All these features differentiate Test PASS from the tools most often associated with CIT such as Jenkins, Bamboo, TeamCity. These tools are commonly used in less safety critical industries and while they are useful for avionics developments, Test PASS fills the gaps.

5. APPLICATION OF TEST PASS TO HUMS

Recall the challenges for HUMS IV&V listed in Table 2-1. In section 4 we have shown that Test PASS is a capable tool with the ability to help solve these challenges. However, the designers of Test PASS realise that in avionics project environment, where pressures of budget and time are normally high, engineers don't necessarily need to invest effort in mastering yet another tool. Hence, the Test PASS team offered their service alongside the tool.

People in the project were allocated to groups, following a concept of responsibility division described in section 4.1.2. An engineer was designated as the Test PASS Integrator. In this role they visited the Southampton site and held discussions with Technical Leaders to understand both the GE HUMS product characteristics and testing equipment involved.

Based on gathered information a draft of the GE HUMS specific test API was proposed. Technical leaders approved proposed interfaces and configuration of environment. From that moment all engineers taking part in the project were able to work in parallel: Software Engineers continued implementation of HUMS software, Testers composed test scripts using only approved API functions, and integrators implemented API functions. This process is summarised in Figure 8.

A "Remote Access" functionality was critical to the enabling the multinational collaboration and allows a constrained resource to be shared between multiple users and as more testers became involved in the project, Test PASS will implement user queuing. Practical considerations, such as the provision of a webcam (Figure 9) also needed to be made. For the Integrator working remotely from Poland, the web cam was a great support while implementing API functions and allowed remote users to inspect hardware settings.

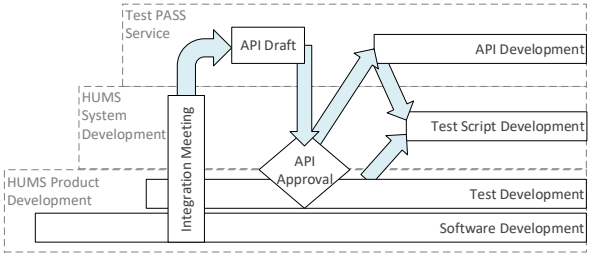

Figure 8: Test PASS Service Concept

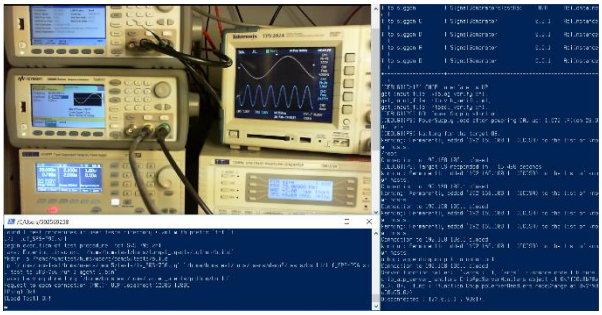

Figure 9: Test PASS Remote Access

A data driven architecture has been developed that allows for multiple test procedures to be organized and automatically orchestrated from a single test initiation. This approach allows new procedures to be easily appended to the test set to keep in sync step with the development. Numerous outputs such as logs and data files are created through the use of the test APIs and the results are automatcally organized by Test PASS based on the test procedure reference. This data management allows for efficient post test review of data to support rapid fault rectification in the event of a test failure.

Running the first test procedures resulted in a long list of failed checks. Far from being disappointing this quick result was beneficial. HUMS V&V engineers could confirm that requirements are testable and integrators were gradually delivering the testing functionality, looking at priorities agreed together with technical leaders. When test procedure was in place, the software team could quickly verify their code by running the test. This greatly speeds up implementation process and is a direct example of the adoption of the Agile principles listed in section 3.

Test PASS is a key enabler for implementing a CIT approach for HUMS development. However, as discussed in section 3, while automated testing is critical CIT requires broader process changes. Figure 10 shows the extensions to a standard CIT approach (Figure 2) that were necessary. After verifying that the software builds from the code repository using Jenkins[3] the software is automatically loaded to the target using Test PASS and a set of basic tests involving around 15 checkpoints are run to verify that the system enters its

[3] https://jenkins.io/

Presented at 45th European Rotorcraft Forum, Warsaw, Poland, 17-20 September, 2019
This work is licensed under the Creative Commons Attribution International License (CC BY). Copyright © 2019 by author(s).

normal operating mode and can acquire vibration data when demanded. This process takes around 5 minutes and typically occurs 2 to 3 times a day. Any failures are immediately rectified by the team. As the development progresses the scope of the tests that are automatically executed increases in line with the capability of the software.

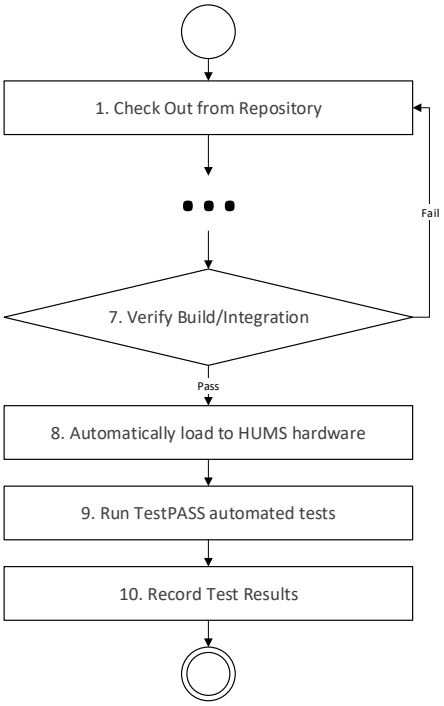

Figure 10: Extension to CIT development steps for HUMS automated testing (cf. Figure 2)

This approach has meant that hardware integration problems have been identified much earlier in the development cycle. It also meant that a detailed understanding of the system start-up and mode transition behavior was required which ultimately led to the removal of extraneous mode transitions, improving the overall performance of the system. This was a direct result of systems engineers becoming more closely involved with the behavior of the software through the use of Test PASS

The Test PASS API is a Python library with simple API calls that are easily integrated with the Python test procedure scripts. Python is a relatively simple scripting language and so systems engineers were able to capture their knowledge in the test scripts and tailor the testing to their needs. The test script is fashioned as a set of off the shelf modules removing the need to duplicate script within the test framework. The main test routine contains a single user defined block of around 15 lines of code defining which acquisition demands, regimes and signals to include in the tests. From this block the automated tests can be configured to run everything from a single bearing acquisition to a full transmission scan.

The results of a transmission scan are a set of CIs for the monitored components. Typically, for a new HUMS installation, CI responses are characterized before fixed alerting thresholds are set. The CI response is a combination of the characteristics of the monitored component, the HUMS installation (e.g. accelerometer location) and the HUMS processing. However, as an FFF replacement it is important that GE's next generation HUMS reproduces the response of the previous generation, within an acceptable tolerance. No characterization should be needed and this represents a high bar for a HUMS product to achieve. Hence, an automated method of verifying the CI response was needed.

First a set of CI results were captured in a results database by exciting the previous generation HUMS with a representative set of vibration inputs (Figure 11). Alongside the results in the results database a set of tolerances were defined.

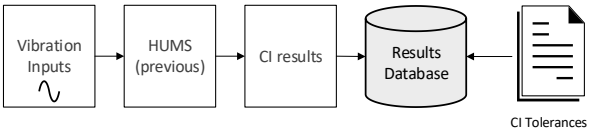

Figure 11: Generation of the results database

Next an automated method of retrieving the expected results was required (Figure 12). A results generator was created that examined the regimes and demands defined in the test profile and parsed the HUMS configuration to retrieve the expected results for a test profile from the results database. The expected results are captured in a CSV file that contains XPath references to the test results XML file. Note that only the results relevant to the specific HUMS configuration and test profile executed are included in the expected results. Expected results can be generated at runtime or in advance of a test run.

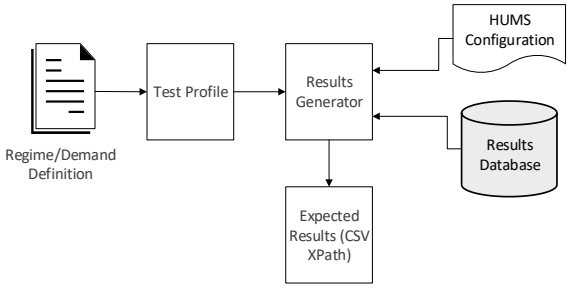

Figure 12: Automatic CI result retreival.

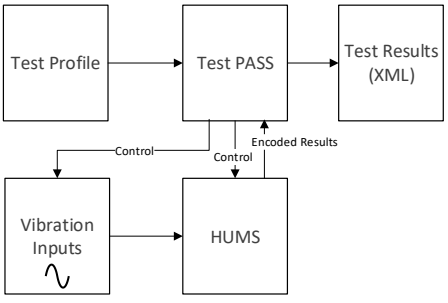

Figure 13: Execution of a test profile via Test PASS

Presented at 45th European Rotorcraft Forum, Warsaw, Poland, 17-20 September, 2019
This work is licensed under the Creative Commons Attribution International License (CC BY). Copyright © 2019 by author(s).

Upon execution of the test profile in the HUMS Test PASS set up (Figure 13), the same representative set of vibration inputs are used to excite the next generation HUMS and produce a set of CI results. For a given test profile Test PASS generates the control inputs required by the HUMS unit and commands the signal generators. The CI results are stored in a proprietary format and Test PASS automatically executes existing HUMS support software to convert results into XML once the HUMS processing has completed.

Finally, once the system operational sequence has been completed, the Test PASS verifier compares the XML results output data with the expected result content. The verification process automatically determines the pass or fail result accounting for tolerances built into the expected results.

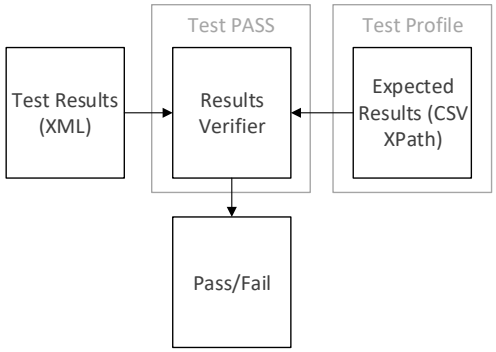

Figure 14: Automatic Results Verification

Each of the processes captured in Figure 12, Figure 13 and Figure 14 have been orchestrated such that they run automatically. It is also feasible to automate generation of the results database (Figure 11) if this process had to be repeated. A change in the regimes or demand captured in the test profile or in the HUMS configuration automatically modifies the tests that are run and the pass/fail determinations that are made. This is important as it automates configuration control of the test set up and means that developers do not need to update tests as development progresses.

The expected values contained in the results database allows the verification of approximately 2.5 million test points. It was conservatively estimated that the manual creation of these test points would take approximately 15 engineers working full time for a year. This does not account for work to update them on a configuration change. With the results generator creation of the expected results set takes around 30 seconds. Hence, the scope of testing that can be economically achieved is dramatically larger and by implementing CIT we have managed to greatly improve test coverage for the GE's next generation HUMS and in the future these principles can be extended to other areas.

The implementation of Test PASS has improved this existing project, but also presents opportunities for further work. Two examples are:

1) **Return to service or acceptance testing**. A typical end to end test run can be executed in less than an hour. As a result, the Test PASS tool and the full suite of automated tests could be used for both acceptance tests on newly manufactured systems or by providing "Return To Service" tests for units that have been repaired.
2) **Aircraft Level Validation.** Raw vibration signals collected from an installation can be replayed through the Test PASS framework and compared against a previous installation. Hence, the influence of a specific installation on a specific aircraft can be quantified and on aircraft installation issues can be quickly resolved.

6. SUMMARY

In summary let's return the challenges for HUMS IV&V presented in section 2.2. Table 6-1 lists these challenges and summarises how they have been resolved by adopting a CIT approach enabled by Test PASS.

Table 6-1: Resolution of HUMS IV&V challenges

Id	Challenge	Resolution
1.	HUMS V&V is a time consuming, manual task.	The adoption of Test PASS and the implementation of additional verification tooling such as the results verifier has resulted in efficiencies and increased the practicable scope of verification.
2.	Traditional HUMS integration methods can make fault localisation difficult	Increased testing at a lower level and daily integration of every build with the target hardware has identified low level integration faults early in the development lifecycle. System engineers have also been much more closely involved with the software development as evidenced by the elimination of redundant mode transitions.
3.	An expectation of improved ROI creates development cost pressures.	CIT for HUMS as described herein provides a way to defeat the "iron triangle". For a fixed scope quality can be improved, unexpected costs avoided and development schedules compressed.
4.	HUMS IV&V is a specialist task requiring specialist resources	HUMS IV&V remains a specialist task, but by capturing the specialist knowledge in the Test PASS framework the dependency on specialist resources (both human and physical) is reduced. The capabilities of the Test PASS framework have made this approach possible.

Presented at 45th European Rotorcraft Forum, Warsaw, Poland, 17-20 September, 2019
This work is licensed under the Creative Commons Attribution International License (CC BY). Copyright © 2019 by author(s).

Id	Challenge	Resolution
5.	As an FFF replacement there is an expectation that the system works first-time.	2.5 million test points can be automatically checked hourly. This greatly increases the scope of testing that is feasible and therefore the robustness of the system. As with any introduction to service there is the possibility of unexpected emergent behaviour, but the likelihood of this is reduced.
6.	Opportunities for on aircraft validation are reduced.	Automated testing makes it feasible to run tests across a wide range of scenarios an can be used to recreate a range of expected or unusual in service conditions. The need for on aircraft experience is reduced as a more representative simulation of the real world is possible.

7. CONCLUSIONS

The safety benefits for Rotorcraft HUMS are proven, however, unlocking the economic benefits of HUMS data requires that unexpected costs in the development and operation of HUMS are minimized. The novel application of CIT best practice, as presented herein, is another tool in the armory of a HUMS supplier such as GE and provides a way to provide improved quality for any given cost. This is favorable to reducing the functionality of a system to meet cost targets.

GE's next generation HUMS represents a step change in capability while still offering FFF backwards compatibility for existing customers. In the past HUMS was largely verified at the system level with a high level of on aircraft testing. Coupled with a Vee lifecycle and global integration approach meant that errors were found, but typically when their resolution was more costly. This problem is not unique and has been addressed by CIT methodologies in the software development arena. These methodologies have been adopted for avionics development but in the author's experience point solutions are typically employed to overcome the challenges of testing safety critical systems and specific functionality of each system.

Test PASS provides a generic method to test safety critical systems and implement and end to end automated testing approach. Critical to its successful implementation has been the support service offered by the team in Warsaw which designed a solution for the system under test and helps the client understand how to best employ the tools at their disposal, without a significant training burden. By coupling this with their own CIT tooling, the HUMS development team have been able to automate over 70% of HUMS tests resulting in efficiencies for the current project even accounting for the costs of setting up the automated test infrastructure. This efficiency is beneficial, but the increased robustness offered by increasing the scope of testing, being able to automatically generate 2.5 million test points in 30 seconds (as opposed to 15 man-years) and then test these points hourly provides much greater confidence in the system. The opportunities for re-using the HUMS CIT infrastructure for other, related applications exists such as return to service testing and aircraft level validation are expected to yield future benefits.

The following conclusions may be drawn from the application of Test PASS to HUMS development.

1. CIT principles, while valid, require careful planning for avionic safety critical systems. HUMS and particularly GE's next generation system with its unique capabilities presents another layer of complexity that must be overcome.
2. The Test PASS framework extends the capabilities of industry standard automation tools to enable automated testing of safety critical hardware. Supported by a service it does this in flexible a way that means that it is suitable for a wide range of avionics applications and avoids point solutions.
3. The Test PASS framework has proven to be a strong foundation upon which to implement an automated testing environment for HUMS. The augmentation of the Test PASS framework with additional tooling has unlocked CIT benefits for HUMS.
4. The collaboration between UK and Polish team members is fruitful because of the one team attitude that exists. Building HUMS specific application knowledge in the EDC has been to the mutual benefit of both teams.

Point 4 is important and has only been eluded to thus far. We have presented the technical aspects of the collaboration between the teams in Warsaw and Southampton, but the human aspects were critical to its success. GE has chosen to build HUMS expertise in Poland and leverage the legacy of HUMS development at the Southampton site. Not only has Test PASS development been conducted in Warsaw, HUMS software development is also ongoing at EDC which leads to the cross-pollination of ideas. Both sites have been open to learning new skills and as HUMS knowledge is growing in Warsaw, Southampton continues to improve its automated testing approach. Rather than treating the other as a supplier or customer we have chosen to succeed or fail as one. The collaboration has set a strong foundation for the future and we look forward to advancing HUMS together.

Presented at 45th European Rotorcraft Forum, Warsaw, Poland, 17-20 September, 2019
This work is licensed under the Creative Commons Attribution International License (CC BY). Copyright © 2019 by author(s).

8. ACRONYMS AND ABBREVIATIONS

API	Application Programming Interface
CAA	Civil Aviation Authority
CAN	Controller Area Network
CCU	Cockpit Control Unit
CI	Condition Indicator
CIT	Continuous Integration and Test
CVFDR	Cockpit Voice and Flight Data Recorder
DAL	Design Assurance Level
DAPU	Data Acquisition and Processing Unit
DFDU	Digital Flight Data Unit
DST	Defense Science and Technology
EDC	Engineering Design Centre
FFF	Form, Fit and Function
GECP	General Electric Company Polska
HMI	Human Machine Interface
HUMS	Health & Usage Monitoring System
I/O	Input/Output
ICD	Interface Control Document
INCOSE	International Council on Systems Engineering
IV&V	Integration, Verification and Validation
LRU	Line Replaceable Unit
MPGS	Multi-Platform Ground Station
OBT	Optical Blade Tracker
OCR	Optical Character Recognition
R&D	Research and Development
ROI	Return On Investment
RTB	Rotor Track and Balance
Test PASS	Test Platform for Safety-critical Systems
UUT	Unit Under Test
V&V	Verification and Validation
WIA	Warsaw Institute of Aviation

9. REFERENCES

[1] UK Civil Aviation Authority, “CAP 1145: Safety review of offshore public transport helicopter operations in support of the exploitation of oil and gas,” UK CAA, 2014.

[2] J. McColl, “HUMS in the era of CAA, JAA, EASA,” in *Fourth DTSO International Conference on Health and Usage Monitoring*, Melbourne, 2005.

[3] UK CAA, “CAP 693: ACCEPTABLE MEANS OF COMPLIANCE HELICOPTER HEALTH MONITORING CAA AAD 001-05-99,” 1999.

[4] A. Draper and J. Gourley, “The Operational Benefits of Health & Usage,” *DSTO HUMS 2003,* 17-18 February 2003.

[5] A. Heather, “Evaluating the Benefit of Health & Usage,” *AHS 59th Annual Forum ,* 6-8 May 2003.

[6] A. Heroux-Theriault, “Achieving HUMS Program Financial Benefits – 12 years of day to day HUMS operations on the CH-146 Griffon fleet,” in *AHS 70th Annual Forum*, Montréal, Québec, Canada, 2014.

[7] J. Stecklein, “Error Cost Escalation Through the Project Life Cycle,” NASA Johnson Space Center;, Houston, TX, United States, Jun 19, 2004.

[8] INCOSE, in *INCOSE Systems Engineering Handbook : A Guide for System Life Cycle Processes and Activities*, New York, John Wiley & Sons, Incorporated, 2015, pp. 81-83.

[9] J. Day and P. Harding, “The Status of HUMS for Helicopters within the UK Ministry of Defence,” in *10th Defence Science and Technology (DST)* International Conference on Health and Usage Monitoring Systems*, Melbourne , 2017.

[10] C. larman, Agile & Iterative Development: A Manager's Guide, Addison-Wesley, 2004, pp. 58-62.

[11] “The Agile Manifesto,” [Online]. Available: https://agilemanifesto.org/. [Accessed 24 June 2019].

[12] M. Fowler, “Continuous Integration,” ThoughtWorks, 01 May 2006.

[13] S. Vanderleest and A. Buter, “Escape the waterfall: Agile for aerospace,” in *IEEE/AIAA 28th Digital Avionics Systems Conference*, Orlando, Florida, 2009.

[14] K. Palmer and D. Blake, “How to Help Your Employees Learn from Each Other,” Havard Business Review, 8 Nov. 2018. [Online]. Available: https://hbr.org/2018/11/how-to-help-your-employees-learn-from-each-other. [Accessed 02 07 2019].

Presented at 45th European Rotorcraft Forum, Warsaw, Poland, 17-20 September, 2019

This work is licensed under the Creative Commons Attribution International License (CC BY). Copyright © 2019 by author(s).

[15] E. Raymond, Basics of the Unix Philosophy. The Art of Unix Programming, Addison-Wesley Professional, 2003.

[16] T. Stanislawski, “Automated display testing in TestPASS,” in *Ada-Europe 24th International Conference on Reliable Software Technologies*, Warsaw, PL, 2019.

[17] SINTEF, “Helicopter Safety Study 2,” 15 December 1999.

[18] RTCA and EUROCAE, “Software Considerations in Airborne Systems and Equipment Certification,” 2011.

[19] SAE International, “ARP4754A: Guidelines for Development of Civil Aircraft and Systems,” 2010.

Research works conducted as part of the project:

An innovative tool for automated software testing for safety critical systems in aviation as a confirmation of the competences of Polish scientific staff of WIA on the international arena

FINANCING AGREEMENT WITH THE NCR&D POIR.04.01.04-00-0121/16 from 13.06.2017

Measure 4.1. "Research and development work " Sub-Measure 4.1.4 “Application Projects” of the Smart Growth 2014-2020 Operational Programme

CPV: 73000000-2 (Research and development services and related consultancy services)

Presented at 45th European Rotorcraft Forum, Warsaw, Poland, 17-20 September, 2019

This work is licensed under the Creative Commons Attribution International License (CC BY). Copyright © 2019 by author(s).