GCRL2000 Research Project: Scoping a FRACAS for UWA Motorsport

Erwin Bauernschmitt
UWA System Health Lab

July 9, 2023

Contents

1 Outline

1.1 Organisation Context

The University of Western Australia Motorsport club(henceforth UWAM or the club) is a student-led club that attempts to compete annually in the FSAE-Australasia student design competition. This competition involves designing, building and racing a formula-style racecar. UWAM offers students a great opportunity to gain practical experience in areas such as engineering design, fabrication, project financing, and team management. The club was founded in 2001 and has been building electric cars since 2017.

There is a small organisational hierarchy with key roles including the Project Manager and Team Leads reporting to the Project Manager. There are seven teams: Electrical, Powertrain, Chassis, Aerodynamics/Cooling, Vehicle Dynamics, Business, and Race/Testing. As evidenced by the names, the teams are mainly formed around technical subsystems of the vehicle, with the Team Leads usually being the most experienced members of the club in the design and fabrication of their subsystem. The rest of the club's members are either general members of a team and report to their particular Team Lead, or fill other specific positions such as the Lab Safety Officer.

New members are introduced to the club and then join a team. The intake of new members is mainly biannual - at the beginning of each semester, following club expo events. However, members can also join throughout the rest of the year. As the club follows an annual competition cycle, it also chooses its Project Manager and Team Leads annually.

As of the 2023 Annual General Meeting, UWAM had approximately 35 active members. All positions at UWAM are unpaid volunteer positions, and members choose how much time to dedicate to UWAM activities. As the club's members are current students, the average membership duration is short, since memberships are generally limited by degree duration. Further, many members may not join in their first year.

1.2 Problem

From the perspective of UWAM being an educational opportunity, UWAM is inefficient at transferring experience and technical knowledge from competent members such as current Team Leads to new members. This issue is due to factors such as experienced members having limited time, a lack of structured new member training, incomplete or missing design documentation, a lack of historical documentation, a poor filing structure, and a lack of high-level subsystem breakdown and explanation. Further, there is a large gap between the competence of new members and the competence required to understand the documentation and processes that do exist, as well as to understand the designs themselves.

Secondly, without a proper system for knowledge management and transfer, there is substantial experience and technical knowledge that is entirely lost from the club when members move on from UWAM. Future UWAM members might be able to contact such past members for help, but without documentation or processes in place, members might not even know that a particular past member could be helpful.

Jointly, these two issues impede learning as members need to inefficiently rediscover previously achieved solutions to their current problems. This is an inefficient allocation of UWAM's human capital and impacts UWAM's competition prospects. The inefficient use of time by new members combined with the additional time load on experienced members due to poor knowledge management practices slows UWAM's design evolution from year to year. Further, it slows the training-up of new members to the point where they are capable of independently managing a design project and training others.

1.3 Project Goal

This project aims to research, define the requirements of, design, build, and implement a Failure, Reporting, Analysis and Corrective Action System (FRACAS) for UWAM. This system will serve as an element of a greater knowledge management and transfer system, allowing current and future members to see records of past failures and how they were dealt with.

The FRACAS will structure the collection of data to be analysed over time, which can reveal annually repeating failures, the bottleneck failures most likely to impact the project schedule, the failures most likely to impact the project budget, and failures that lead to cascading further failures. The intention is that this knowledge capture system can improve UWAM's scheduling, budgeting, management, vehicle testing, and the focus of future design efforts. It is a system that has the potential to become more valuable over time.

1.4 How will we achieve this?

FRACAS reporting is widely used in industry. There are existing processes and documentation that can be leveraged by the team. The key challenges are in the training of team members, the discipline required to enter data, and the willingness of people to use the data. There needs to be a shared responsibility for maintaining the system and one person needs to be made accountable for ensuring it is updated and used.

Because of the organisational and behavioural aspects of this project, research and consultation with UWAM, the project mentor, and experienced programmers will continue throughout the duration of the project. The system requirements will be defined Using principles and practices from systems engineering, following research findings, and consultation with members of UWAM. This will include requirements for the implementation of the system. A system that meets these requirements should then be conceptually designed. After exploration of different build options, the design should be built, tested against the system requirements and modified if the build falls short of the requirements.

1.5 What will we deliver at the end?

The aim is to deliver the implementation of a Failure Recording, Analysis and Corrective Action System for UWAM, along with a presentation on the system and a report detailing the project.

2 FRACAS

2.1 History

In 1980, the US Government Department of Defence published the MIL-STD-785 Military Standard entitled Reliability Program for Systems and Equipment Development and Production [1]. It defined several components of a comprehensive reliability program, of which one component was a Failure Reporting, Analysis and Corrective Action System (FRACAS).

In 1985, the Department of Defence published MIL-STD-2155 Failure Reporting, Analysis and Corrective Action System [2]. This standard established the requirements and acceptance criteria of such a FRACAS for the US Navy and all Department of Defence agencies.

2.2 Users of FRACAS

The standards were approved for public release, and companies in the defence and aerospace industries began to integrate FRACAS into their reliability programs. As it is a requirement from the Department of Defence, companies contracted for the development of software, systems or equipment for the US military are required to use FRACAS. These include companies such as Lockheed Martin, Boeing, and BAE Systems. It is also used by the military branches of other governments, such as the UK Ministry of Defence.

FRACAS is used much more broadly in other industries that produce machinery or software requiring a high degree of reliability, such as the automotive industry. It is even a requirement as part of certain FDA certifications. ISO 9001 outlines requirements for the definition and use of a FRACAS or Corrective And Preventative Action (CAPA) system, and there are over one million organizations across 170 countries that are certified to the standard. Thus, it would be valuable for UWAM members to gain experience in the development, establishment and use of a FRACAS.

2.3 Purpose

In its original inception, FRACAS was a closed-loop system for the formal recording of failure incidents. The causes of recorded failures could be analysed and subsequently, timely adjustments could be made to designs to reduce the likelihood of the same failure recurring. Further, with a better understanding of a design's likely failures, the designs could also be adjusted to simplify or reduce maintenance tasks.

Apart from the benefit of improving the design of a particular system, software or piece of equipment, a FRACAS also serves as a historical record that can inform similar future designs. In this respect, it is a system that becomes increasingly useful over time. With regards to UWAM, a FRACAS has the potential to improve current designs, as well as improve the rate of evolution from one year's design to the next.

As UWAM is also an opportunity for students to develop themselves and gain practical experience, the structure of a FRACAS teaches good problem resolution for engineers, and the record of past failures provides teaching content for common problems and solutions.

2.4 Useful Outputs and Features

A FRACAS can be built to run an automatic analysis of logged failures, and then output the results of such analysis as useful metrics. However, which outputs are useful depends on the purpose of the system.

Common outputs of FRACAS systems are Mean Time Between Failure (MTBF) estimates, Mean Time to Repair (MTTR) estimates, and the ability to group and view failures in various ways such as by failure type, time, location, system or part type.

Other outputs might include status reports on unresolved failures or summary reports on past failures for the purpose of review or validation.

2.5 Types of Failure

Although more commonly used for software and physical equipment, a failure can occur in any system. A system can be defined as anything that has components that work collectively to perform a specified function. It is obvious that a physically built car can be considered a system, as it is made of physical components and has a specified function that is racing. However, the UWAM members can be considered the key components

of a system that produces an annual car design. Further, the UWAM members are also key components of the system that manufactures and tests a functioning car annually.

Just like the physical components of a car, the team members have required functions that they may fail to execute at the specified level of performance, which is a failure that impairs the greater system from performing its function. Thus, a FRACAS can be used to record and resolve all sorts of failures in the design phase, the manufacturing phase, the testing phase and the racing phase of the annual build cycle. For example:

- Physical failure of a component on the car during racing.
- Failure of electronic hardware during testing.
- Missing the target number of test hours or driver training hours.
- Failure to follow a standard design convention.
- A major setback in the project timeline.
- Failure to budget for a major expense.

2.6 Information to Record

Typical pieces of information included in a FRACAS failure record are:

- Record creator and contact.
- Record owner and contact.
- Failure description.
- Failure time and location.
- Failure category/subcategory.
- Description of failure.
- Response action taken.
- Corrective action plan.
- Date of resolution (if resolved).

2.7 Structured VS Unstructured Info

A FRACAS provides a record of useful data in addition to being a closed-loop system for failure resolution. That data can be analysed in various ways to provide design and reliability insights. However, some types of data are easier to analyse than others. Structured data fields in a record such as time, location, and owner name can be readily processed or visualised. However, it is much harder to extract useful information out of unstructured data such as written descriptions, titles, and corrective actions when analysing a large set of records, potentially requiring advanced tools such as natural language processing. Although unstructured data fields offer more flexibility when recording useful information, this must be balanced against ease of analysis. Consultation with the future users of the system is necessary for tailoring data fields.

2.8 Existing Options

The first systems would have been developed in-house by companies required to comply with the US government standards, but today there are FRACAS options on the market.

XFRACAS is a web-based system developed by ReliaSoft that purports to be highly customisable and is meant to integrate with other software products offered by ReliaSoft for purposes such as FMEA, system reliability analysis, and life data analysis. It is based on the .NET Framework and does not involve a client installation, being accessible from a web browser. It includes support for user profiles, group-based problem solving (via various standardised procedures), personalised user portals, serial number tracking, reports, data visualisations, and data exports. A link to their product information document can be found by clicking **here.**

2.9 Methods of Implementation

There are multiple ways a FRACAS could potentially be implemented, such as the following:

- Excel spreadsheet (with Macros potentially)
- Web-based application
- Mobile application
- Desktop application
- Raw database (such as an SQL server)

2.10 Common Issues

Relex Software Corporation is a company that offers advanced reliability engineering software. The company released an article entitled "Best Practices for a FRACAS Implementation" that identifies three key organisational issues that limit the effectiveness of a FRACAS when it is designed and implemented [3]. The first issue is that a complex organisational structure with many divisions and many hierarchical levels results in too many stakeholders being involved and compromises the timeliness of failure resolution. The second issue is that poor definition and prioritisation of the system's goals can lead to the system failing to meet expectations once implemented. The third issue is that an unclear form or unnecessarily long form can be too confusing for users to complete or can leave users feeling the process is too time-consuming, resulting in inaccurate records or reduced usage of the system.

An article from BMT Reliability Consultants entitled "Failure Recording, Analysis and Corrective Action System - Good Practice" identifies several other organisational issues [4]. These issues include a lack of feedback to the originator of the failure record, a lack of data collected for root cause analysis, poor data tracking for trend identification, and poor distribution of data and results to other interested parties in the organisation.

Aerospace and defence form AAI Corporation released an article entitled "Increasing the Effectiveness of FRACAS" that identifies some more technical issues that occur when using a FRACAS, particularly when the system's design does not match the scale of the application [5]. The issues identified include building a system that can use test data in different formats for automatic analysis, and a lack of integration between design engineers and those responsible for analysing and reviewing failures. The article also raises the issue that originators of failure records can under-utilise fields for free text entry and can fail to record available data that could be useful for analysis outside of determining the failure cause and mechanism.

2.11 Guidelines and Best Practices

MIL-STD-2155 is the inception of the formalised concept of a FRACAS, offering the first set of requirements and the first guidelines for design and implementation [2]. Furthermore, articles [3, 5, 4] all offer guidelines for resolving the issues they identify, as well as more general guidelines. These guidelines and best practices have been grouped below into the phases of planning, designing and implementing a FRACAS system.

Planning:

- Consult users of the system to tailor requirements and data fields.
- Determine what is essential data to eliminate unnecessary recording.
- Design a full workflow process.

Design:

- Use multiple-choice fields to better characterise failures and ease recording.
- Include an organisation breakdown in the form.
- Tailor presented information to user role context and their tasks.
- Have a free text log entry that each stakeholder can use.
- Use a web-based system for feedback, visibility and faster processing.
- Create a visual flow diagram of the workflow process for users.

Implementation:

- Review process for validating completeness, accuracy and timeliness.
- Ensure feedback is provided to the originator.
- Evolve the system by reviewing functionality and needs.
- Validate changes to the system by reviewing their effects.
- Evolve the workflow process alongside the system.
- Produce a guide of key principles and a process for evolving the system.

3 System Requirements

3.1 UWAM Interviews

In accordance with the best practices outlined in [3], brief meetings were held with key members of UWAM to determine what were the most important desired outcomes for a FRACAS. There are many ways a FRACAS might benefit an organisation, including the following:

- Improving reliability
- Reducing cost
- Ensuring timely resolution
- Providing management visibility

- Knowledge management and transfer
- Improving next-gen design
- Providing assurance
- Training good practices

Meetings were held with the Electrical Low Voltage Lead (Nathan Mayhew), as well as the current (Jess Gugliotta) and previous (Julian Blair) Project Managers. The objective of these meetings was to guide the definition of the system requirements to help ensure that the system will be sufficient to achieve its goals without becoming too unwieldy and complicated.

An explanation was given about what a FRACAS is, as well as who uses a FRACAS and why organisations might use one. A broad range of possible failures at UWAM was also given to contextualise the use of a FRACAS at UWAM. Once the presentation was complete, a discussion was opened about what was the most important reason for using a FRACAS in the context of UWAM. The meetings ended with a general discussion about issues that might arise in implementation and features that might be useful or improve the ease of using a FRACAS.

All three members independently identified improving knowledge management and transfer to be the most important goal. The second most important goal was improving the timeliness of resolutions via a closed-loop process, followed by improving management visibility and team communication.

These goals guided the definition of the requirements in the subsequent sections, and the discussions led to requirements being defined for several additional features.

3.2 Functional Requirements

These requirements set out what features the system must have to perform its desired function. They are divided into member functional requirements (features that all members would use) and team lead functional requirements (features that allow team leads to manage their team members).

3.2.1 Member Functional Requirements

Identifier	Name	Description
FR01	Failure Recording	The system shall record information about failures, their causes, their mechanisms, the corrective actions implemented, and the effectiveness of the actions. This is to perform the core function of a FRACAS.
FR02	New from Duplicate	The system should allow new failure records to be created by duplicating existing records and modifying them before saving. This could accelerate the workflow for the recording of failures similar to past failures.
FR03	User Profiles	The system shall require members to log into user profiles from a login portal. This will permit access to the system, allow for failure ownership and provide contact details.

FR04	Profile Creation	The system shall allow people to create new accounts from the login portal and request to join an area team in the organisation.
FR05	Personal Dashboards	The system shall present users with a personalised dashboard upon logging in, showing their unresolved failures, their past resolved failures, their assigned learning failures, and their bookmarked failures, as well as a notifications panel.
FR06	Commenting	The system shall support comment threads on failures. This is for feedback and collaborative problem-solving.
FR07	Partial Completion	The system shall allow incomplete failure records to be saved for later completion or deleted. This will reduce work being redone and provides flexibility.
FR08	Status	The system shall display the status (Incomplete, Complete, Validated) of a failure's record section, analysis section, and corrective action section. The system shall also display the status of the solution (Not Implemented, Implemented, Reviewed).
FR09	Owner Notifications	The system shall notify failure owners of assigned failures, assigned due dates, upcoming due dates, all comments, edits by management, deletion, upcoming validation dates, and validation outcomes.
FR10	View Past Failures	The system shall allow all users to view all past and current failure records. This will enable learning from past failures and the investigation of failure trends.
FR11	Filtered Searching	The system shall allow filtered searching of failure records by time period, competition year, technical team/vehicle subsystem, failure owner, and keywords. This will enable learning from past failures and the investigation of failure trends.
FR12	Bookmarking	The system shall allow all users to bookmark past failure records, which should be visible on their personalised dashboards for future reference.
FR13	Team Dashboards	The system shall have dashboards displaying unresolved failures, corrective actions for failures needing validation, and analysis results of failure trends and statistics. This will improve team-wide visibility and will enable communication and scheduling.
FR14	Report Generation	The system shall be capable of generating a report of all failure records for individual projects, or for selected failure records. This will produce a deliverable document for compliance and assurance purposes.

3.2.2 Team Lead Functional Requirements

Identifier	Name	Description
FR15	Teams Management	The system shall allow team leads or management to create new teams, appoint team leads, view team members, and accept members requesting to join teams.
FR16	Validation	The system shall require team leads to validate the completeness/accuracy/adequacy of the failure records. This validation should occur in three stages: the failure record section, the analysis section, and the corrective action section. The validation should be a simple yes/no but should also allow for comments if necessary.
FR17	Review	The system shall require team leads to review a completed failure record for the purpose of sharing lessons learned, integrating with planning for the future, and validating the effectiveness of the corrective action that was implemented. The review should be a simple yes/no but should also allow for comments if necessary.
FR18	Management Dashboards	The system shall present team leads and management with a second personalised dashboard showing the active failure records of their team members, as well as a section for failures requiring validation of completeness/accuracy/adequacy.
FR19	Assigning Ownership	The system shall allow management to assign failures to team members. This will allow for workload allocation and management.
FR20	Due Dates	The system shall allow the setting of due dates for unresolved failures. This will ensure timely completion and integration with team scheduling.
FR21	Editing Records	The system shall allow management to edit or delete incorrect records. This will allow for the validation of record completeness and accuracy.
FR22	Review Dates	The system shall allow the setting of review dates to assess the effectiveness of corrective actions. This will ensure validation occurs and allows for integration with team scheduling.
FR23	Manager Notifications	The system shall notify management of newly created failure records, upcoming due dates, relevant comments, and upcoming validation dates.
FR24	Learning Assignment	The system shall allow management to select past failure records and assign them to members for learning or review. The failure records will be visible on the assignee's personalised dashboard and will be marked as reviewed once read.

3.3 Performance Requirements

These requirements define how well the system must perform its functions.

Identifier	Name	Description
PR01	Aesthetic Design	The system shall be aesthetically pleasing and intuitive to navigate.
PR02	Streamlined Workflow	The workflow process shall be as simple and streamlined as possible.
PR03	One-Click Report	The system shall allow the creation of a failure record with one click from anywhere in the system.
PR04	Auto-Completion	The system shall auto-complete as much of the form as possible, considering the failure owner's role in the organisation as well as past records of the owner (allowing changes if incorrect).
PR05	Structured Fields	The system shall keep as many data fields structured and multi-choice/drop-down as possible.
PR06	Minimum Fields	The system shall record the minimum necessary information required for searching and analysis.

3.4 System Technical Requirements

These requirements define technical requirements relating to the system's software design.

Identifier	Name	Description
SR01	Low Cost	The system shall cost less to develop and implement than purchasing/subscribing to existing market options over a 10-year period.
SR02	Device Compatibility	The system shall be accessible from computers and mobile devices.
SR03	User Count	The system shall be capable of hosting at least 200 user profiles.
SR04	Storage Duration	The system should store records in an accessible way for at least ten years before archiving records.
SR05	Data Output Format	The system shall store or shall be able to output selected data in a format that allows for easy external analysis.
SR06	Security	The system shall prevent people outside of the system from viewing or interacting with communications and data on the system.
SR07	Maintainability	The system shall be built with a strong focus on extensibility and modularity to make adding new features or performing changes easier.

3.5 Implementation Requirements

These requirements define how the system should be implemented once it has been built and tested.

Identifier	Name	Description
IR01	Team Training	Team-wide training shall be performed upon system roll-out.
IR02	Workflow Diagram	The system shall be delivered with a visual workflow process diagram for users.
IR03	Onboarding Integration	Account creation should be integrated into the new member onboarding process.
IR04	Team Meetings	Failure review should be integrated into weekly team meetings.
IR05	Technical Meetings	Lessons learned and validation should be integrated into technical area meetings.
IR06	Annual Review	Annual meetings should be implemented for failure summary and planning for the next year.

References

[1] *MIL-STD-785 Reliability Program for Systems and Equipment Development and Production*, US Department of Defence, 1980.

[2] *MIL-STD-2155 Failure Reporting, Analysis and Corrective Action System*, US Department of Defence, 1985.

[3] E. Hallquist and T. Schick, "Best practices for a fracas implementation," in *Annual Symposium Reliability and Maintainability, 2004 - RAMS*, 2004, pp. 663–667.

[4] O. Markwardt, "Failure recording, analysis and corrective action system—good practice," *Safety and Reliability*, vol. 31, no. 2, pp. 5–8, 2011.

[5] M. Ciemian, "Increasing the effectiveness of fracas," in *2008 Annual Reliability and Maintainability Symposium*, 2008, pp. 59–63.