the newsletter of
RELIABILITY, MAINTAINABILITY & SUPPORTABILITY

June 2017 • Volume 21 • Issue 2

THIS EDITION

Published Quarterly

Ann Marie Neufelder

The State of the Art in Software Reliability

"It's hard enough to find an error in your code when you're looking for it; it's even harder when you've assumed your code is error-free."
—Steve McConnell[1]

1. Introduction

The first recorded software failure was in 1962. The Mariner 1 rocket with a space probe headed for Venus diverted from its intended flight path shortly after launch. The rocket was destroyed less than five-minutes after liftoff. The failure was due to a missing "overbar" in a formula. The handwritten formula was correct but the programmer had missed the averaging bar when writing the code. Without the smoothing function the software treated normal variations in velocity as if they were serious which caused faulty corrections that sent the rocket off course. The cost of the missing superscript or hyphen was $18.5 million in 1962 dollars.[2]

Since 1962, countless system failures have occurred due to software and firmware. In 1962 software sequentially processed and was small enough to be written by one software engineer. In 2017, typical systems are event driven contain millions, and sometimes multi-million lines of code. Hence, the effect of the software and firmware on the reliability can no longer be ignored. If a system flies, moves, launches, shoots, orbits, hoovers, diagnoses, controls, or analyzes it probably has a considerable amount of software and firmware.

Software reliability models have been used for decades but have been slow to acceptance largely due to exceedingly poor guidance from the academic community, resistance by reliability engineers and denial by software engineers.

In September of 2016 the IEEE 1633 Recommended Practices for Software Reliability was revised to address these roadblocks. The 2008 edition was too theoretical and did not adequately cover software FMEAs, early prediction models, or the factors and risks that effect software reliability.

The 2016 edition was developed, reviewed and approved by members of industry from the U.S. Army, U.S. Navy, Missiles Defense Agency, NRC, NASA, medical device manufacturers, defense contractors and several others.

2. The IEEE 1633 Recommended Practices for Software Reliability

The document was written to be "actionable." There are step by step instructions for how to perform every software reliability task, the criticality of that task, and how to tailor the tasks to a given project and how to tailor the tasks to an incremental or agile development model. The document is arranged according to the order that the tasks will be executed.

2.1 Planning for Software Reliability

The very first steps towards reliable software is to characterize the software system, define failures and criticality, perform an initial risk assessment, assess the data collection system and tools, and

1 McConnell, Steve "Code Complete", Microsoft Press, 1993.

2 Parker, P. J., "Spacecraft to Mars", Spaceflight, 12, No. 8, 320-321, Aug. 1970 http://nssdc.gsfc.nasa.gov/nmc/spacecraftDispla y.do?id=MARIN1

Russell A. Vacante, Ph.D.

Turning the Page: Enhancing the RMS Partnership Activities and Capabilities

Since 1993, the RMS Partnership (RMSP) has tracked changes resulting from the emergence of new technologies and the increased sophistication and growth of hardware-software systems. To date, the RMSP has successfully served the defense community with comprehensive coverage of reliability, maintainability and supportability issues within the overarching systems engineering and the logistics framework. Our core objective remains focused upon improving communication, and thereby hopefully also cooperation among professionals as we expand our outreach to include other industries.

The RMSP will continue to publish its quarterly informative newsletter, our semi-annual professional journal, to teach on-site and online short-term training courses, and to provide subject

continued on page 11

finalize the software reliability plan. Software reliability predictions are performed on custom software, Commercial Off The Shelf (COTS) software, reused and auto generated software and firmware. Prior to assessing software reliability there must be a concrete definition of what constitutes a software failure that is specific to the system under analysis. Without concrete definitions, the reliability figures have little meaning.

Before the software is even written, it's possible to assess the low-medium-high risk of the software development. In summary, if anything "new" is happening on a particular release such as new target hardware, new technologies or turnover in personnel which results in "new" people working on the software, the software has a risk. Research has shown that if there are more than two risks on one software release that the project is much more likely to fail than otherwise.[3] In that case, more software reliability activities may be necessary. Hence, the software reliability plan is finalized after it is clear what the risk is.

Another common risk is assuming too much reliability growth. Whenever new features are added to the software or firmware the reliability growth "resets" as shown in Figure 1. One of the most common oversites is assuming "indefinite" reliability growth. An even bigger risk is when the reliability growth is actually decreasing due to defects piling up from release to release.

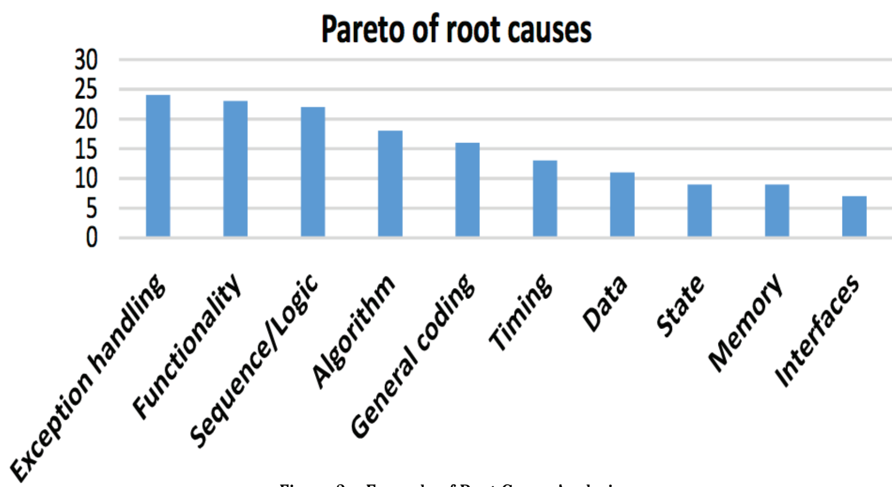


Figure 2 – Example of Root Cause Analysis

2.2 Develop Failure Modes Model

One of the most important qualitative tasks is to identify the types of failure modes to expect and the most likely root causes. There are several dozen software failure modes and more than 400 root causes that apply to every software program. There are several hundred other root causes that apply to certain types of applications such as e-commerce, etc. The 5 most common failure modes are:[4]

1) faulty functionality
2) faulty sequencing
3) faulty timing
4) faulty data
5) faulty error handling.

These failure modes can apply to the requirements, interface design, detailed design, maintenance actions, installation scripts, use cases. In order for a software organization to make any improvements in reliability, the most common failure modes and root causes and the artifacts that are contributing to the most failures need to be identified first. From that the Software/Firmware Failure Modes Effects Analysis can be performed as well as the Fault Tree Analysis.

Note that contrary to popular myth there is no "default" Pareto for software root causes. The most common root causes are unique to every software release. Figure 2 is only an example of a Pareto of root causes and should not be considered as generally applicable to all software projects.

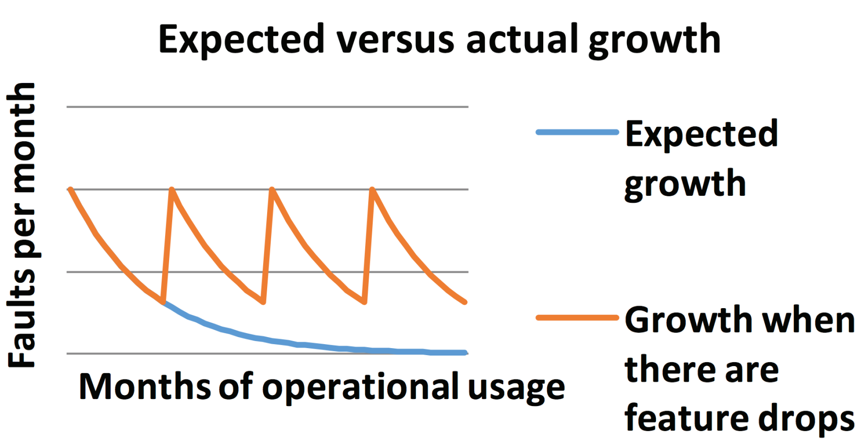


Figure 1 – Expected versus actual reliability growth.

2.3 Predict Software Reliability During Development

3 Neufelder, Ann Marie, "Four things that are almost guaranteed to reduce the reliability of a software intensive system". Huntsville Society of Reliability Engineers RAMS VII Conference, November 4, 2014. Copyright, 2014.

4 Neufelder, Ann Marie, "Effective Application of Software Failure Modes Effects Analysis" A CSIAC State-of-the-Art Report, CSIAC Report Number 519193, Contract FA8075-12-D-0001, Prepared for the Defense Technical Information Center, 2014.

One of the most important software reliability activities is to predict the reliability early in development so as to guide planning and allow for sensitivity analysis. These predictive models have been used since 1987.[5]

The best predictive models depend on:

1) Developed or update recently (in last ten years)
2) Has many sets of data to comprise the model
3) Has several assessment factors that aren't difficult to measure
4) Is not subjective

The simple prediction models have only a few inputs and are consequently less accurate than the detailed assessment models. However, the shorter models are often useful for quick ballpark assessments early in the program. The detailed methods support sensitivity analysis which can identify design related issues that affect reliability. For example, incremental development, when performed correctly, can reduce the effective size of the code as well as the defects subsequently reducing the failure rate. The detailed models can also be used for assessment purposes. Subcontractors and COTS vendors can be chosen based on the assessments.

The predictive models are used to predict:

1) Defect density
2) Remaining defects
3) Failure rate, MTBF, MTBCF
4) Availability
5) Reliability (probability of failure over a specified mission time)

The document provides guidance for how to merge the software predictions with the hardware predictions to yield system reliability predictions.

5 Science Applications International Corporation & Research Triangle Institute, Software Reliability Measurement and Testing Guidebook, Final Technical Report, Contract F30602-86-C-0269, Rome Air Development Center, Griffiss Air Force Base, New York, January 1992.

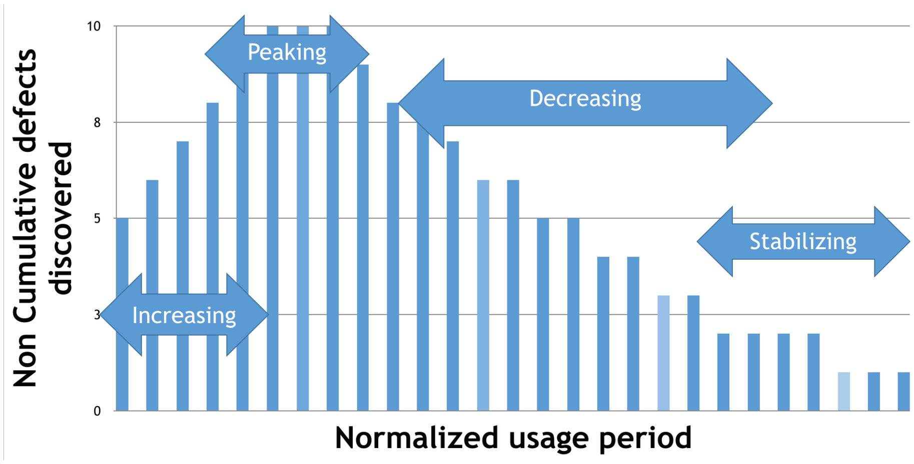


Figure 3 – The Failure Profile of Software

2.4 Software Reliability During Testing
Once the software has been integrated and is being tested from a black box perspective the reliability growth models can be used to compliment the predictive models.

The reliability growth models have been used for decades. Prior to the 2106 IEEE 1633 the existing guidance on these models was far too theoretical. The software failure rate can have one of these possible trends as shown in Figure 3:

1) increasing
2) peaking
3) decreasing
4) stabilizing
5) some combination of increasing and decreasing.

Most of the existing models assume that the failure rate in decreasing. This document provides guidance on what to do if that's not the case as well as how to:

1) Determine which trend is currently applicable
2) Select the model that corresponds to that trend
3) Collect the data and format it
4) Use the models to forecast total defects and future failure rate, provide for adequate warranty staffing, etc.

2.5 Determine a release decision
The risk of deploying software or firmware is related to these 5 things:

1) The amount of black box test coverage (testing from end user or system point of view)
2) The amount of clear box test coverage (covering the lines of code, decisions, etc.)
3) The amount of stress coverage (testing of failure modes, error handling)
4) Whether the current reliability predictions are in line with the required reliability objectives
5) Whether the remaining defects is live-able

Contrary to popular belief, testing is not an "either-or" situation. Just because the software is tested against the requirements does not mean that all of the lines of code have been tested. Typically, black box testing only covers less than half of the lines of code. Testing 100% of the lines of code doesn't guarantee that all stresses and failure modes have been covered as these are usually related to "missing code".

One of the most important advantages of the reliability models is that they indicate an increasing failure rate. If the failure rate is steadily increasing at the time of deployment, the chances of that release being successful are virtually remote.

Lastly, it's possible that a particular release meets all of the release criteria but will result in “defect pileup”. According to research conducted by this author[6] the leading cause of a distressed software release is the previous release.

Defect pileup can leave too many defects to be fixed for future releases. At the very least, it should be predicted to ensure that adequate staffing is provided to ensure that the people developing the code for the next release aren’t interrupted with unscheduled maintenance. Figure 4 illustrates how defects can pile up if they are spaced too closely.

6 Neufelder, Ann Marie, “The Cold Hard Truth About Reliable Software - Version 6f”, 2016, 76 pages.

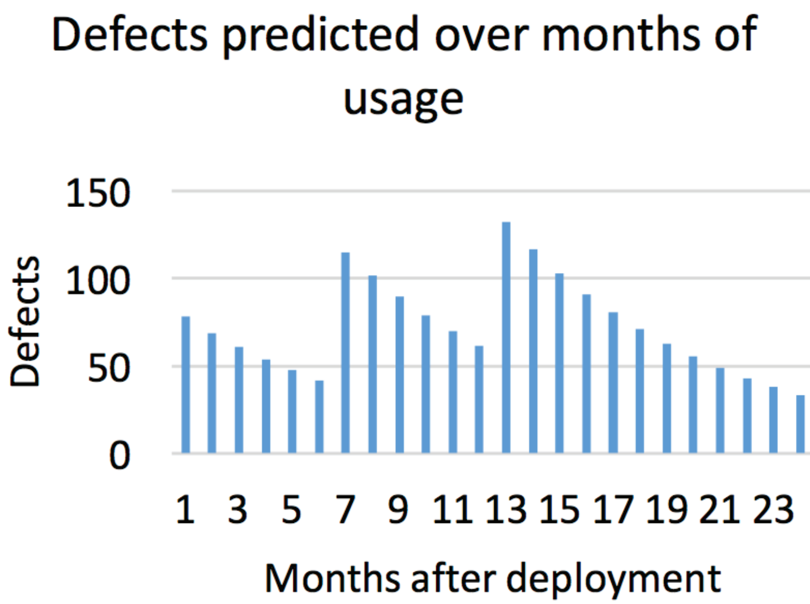


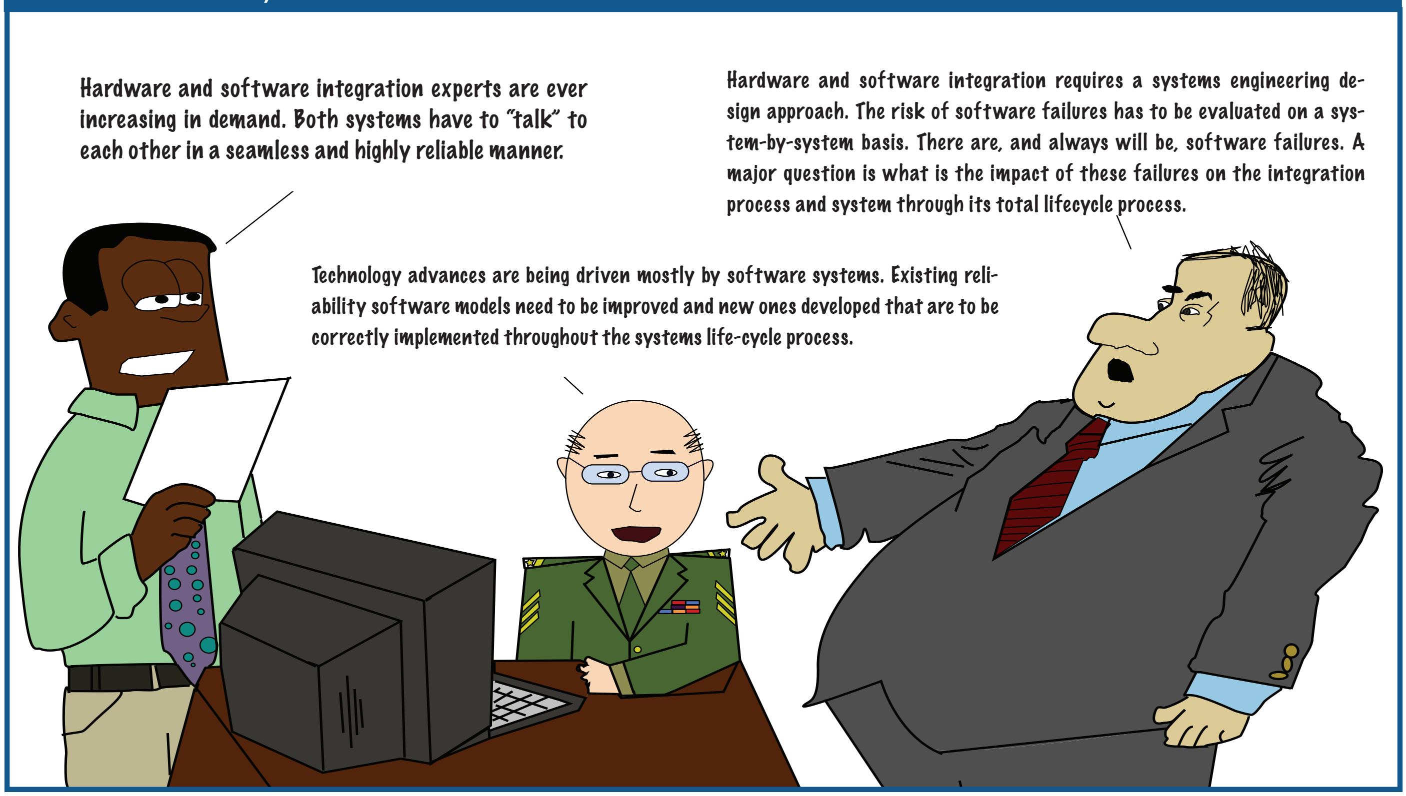
Figure 4 – Example of Defect Pileup

3. Summary

The updated IEEE 1633 Recommended Practices provides an industry approved resource for software reliability planning, failure modes analysis, prediction, and reliability growth estimation and decision support. ●

Another Day At The Office

by Russell A. Vacante, Ph.D.



ONLINE COURSE OFFERING

July 31–November 17, 2017 | $75.00 per course, per student.

The courses listed below can be taken individually or as a collective package.

The module title for the four courses listed below is "Risk Identification and Decision Making in the Internet of Things (IOT) Development." Each course is four weeks long and will require 3 hours of study per weeks to complete.

COURSE 1
Introduction to the Internet of Things, Embedded Systems and Big Data
July 31 – August 25

COURSE 2
Making Decisions – Cost and Technology Basics
August 28 – September 22

COURSE 3
Technology Evaluation and Trade-off Analysis
September 25 – October 20

COURSE 4
Decision Techniques and Tools
October 23 – November 17

Course Registration: *www.rmspartnership.org*
Please direct questions to the course instructor, John Blyler at: (503) 348–1342

QUIZ YOURSELF

An electro migration model for testing (MTTF = AJ-n e Ea/KT)…

a) …is a chemical oxidation-subduction reaction.
b) …requires a layer of water approximately 5 monolayers thick for ECM to occur.
c) …current density to a power and the Arrhenius temperature dependence.
d) …associates ECM on printed circuit board assemblies (PCBSs) as conclusively demonstrated.

…answer on Page 9.

IN MEMORIAM

James Rodenkirch

James Rodenkirch, 72, passed away March 28, 2017, in St. George, Utah after a courageous battle with bone cancer. He and Janet had been married 35 years and have resided in St. George for the past 11 years.

He served as Editor-in-Chief of RMS Partnership professional journal, The Journal of Reliability, Maintainability and Supportability in Systems Engineering, for the past eight years. His friendship, selfless service, and professional contribution will remain his enduring legacy.

He retired as a Navy Chief after 21 years of service. For the next 17 years he worked as a systems engineer in DoD after graduating Magna Cum Laude in Electronics Engineering from Chapman University. Most recently, he completed a Master of Science (M.S.) in Systems Engineering and taught a Complex Systems Architecting course as an Adjunct Lecturer at Southern Methodist University for eight years. Jim was an avid ham radio enthusiast in his spare time.

Jim's RMS Partnership friends and colleagues offer our sincere and deepest condolences to his family. We all benefited from knowing and working with Jim. We will continue to aspire to achieve his high professional standards and humanitarian goodness.

Bon Voyage Jim, we already miss you.

Peter Cook, Lead Mentor-Team 4450, Olympia Robotics Federation

High School Student Robotics Teams Learn Reliability and Maintainability

Team 4450 Mentors and Students

There are almost 5,000 robotics teams around the World where students are learning through hands-on design, build and competition about basic concepts in reliability and maintainability for electro-mechanical machines. These teams are composed of high school students who are organized by a U.S.-based organization called FIRST (For Inspiration and Recognition of Science and Technology) that issues a challenge to all the teams in January of each year.[1] The teams have six weeks to design and build a 120-pound robot that responds to the challenge. Then they go to competitions at the local and state levels, and if successful they advance to the World Championships.

This year the challenge was a game called Steamworks with a theme from the 1800s that featured fuel for "boilers" and gears for "airships". The game is played on a large field with two alliances of 3 robots each. The robots gather up fuel in the form of 5" diameter balls similar to whiffle balls and toss them into a cylindrical container called a boiler. The robots are also challenged to pick up large gears and carry them to an airship where a pilot loads them on board and uses them to spin up rotors. (see game animation at FIRST Steamworks on YouTube). At the end of each 2.5 minute match, the robots are challenged to "board the airship" by climbing a 5' rope that is deployed by the pilot. This is quite a challenge, but all the teams produced different types of robots (with some common control systems) that could address it.

My team, the Olympia Robotics Federation, (see picture) is a good example of a high-performing team. We have 30–40 students from three high schools and 13 mentors from teachers to community volunteers who have different skills from engineers to shop instructors and businessmen. The team trains in the summer and fall to develop the skills needed for the robot design and development work, based on past challenges and also develops business and leadership skills to show to the community what we do and raise money to support the team. The team is similar to an after school sports team and the competitions are run like sports events. (FIRST calls it the Sport of the Mind). The mentors are there not to do the work but to ask the hard questions and make suggestions while the students do 90% of the work.

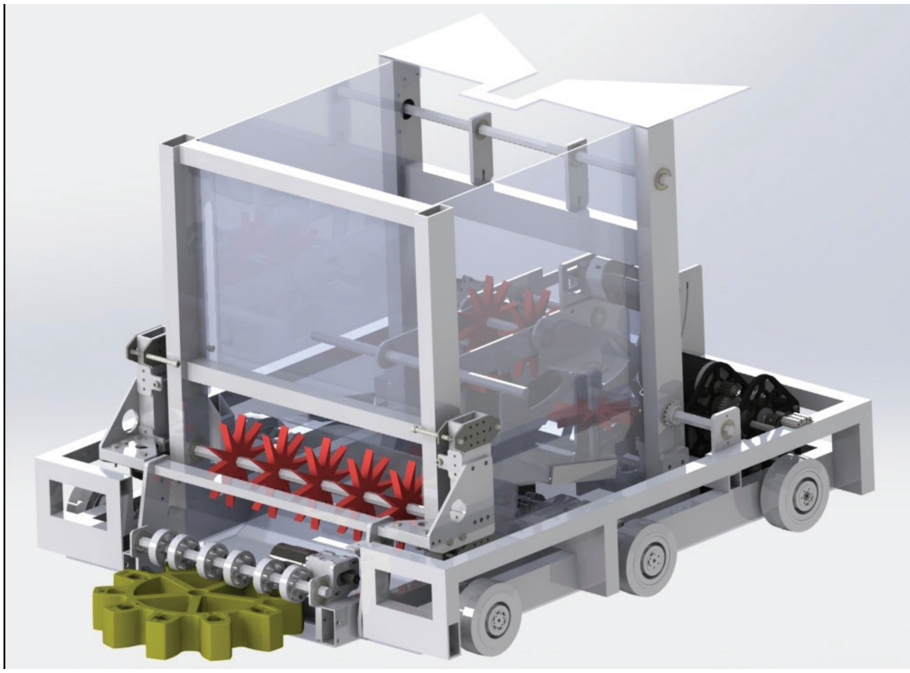
The first week after the challenge, the team develops a strategy to address the challenge and identifies the robot functions that are needed to fit the strategy. The designs must be efficient and the total weight is limited to 120 pounds. The robot dimensions are also constrained to certain limits, which change every year. The team uses CAD software for the design and the robots are primarily made from aluminum which is cut according to the CAD design. (see rendering of this year's robot, the Enterprise). There are many discussions of design features and objectives, including how to carry out multiple tasks with the same components and how to integrate the functions in the space limitations of the robot. Accessibility of key components for maintenance is discussed and calculations of force, torque and power are made for the design (especially the climbing mechanism), but some of the parameters are worked out in testing and trial and error (e.g., shooting of the fuel balls) on a practice field.

The real tests of the robot occur during competition.[2] For example this year we found out in the first competition that out robot could climb well up to the end of the match, when suddenly it failed in mid-climb, but it was not a mechanical failure. We finally traced it to our motors drawing more current than the circuit breakers were designed to handle and if we were still climbing after ten seconds, we could trip the circuit breakers. They would reset, but not before the end of the match. We made a software adjustment to reduce the power level in the climb to 90%. There was a balance to strike between the speed of climb and the current that could be drawn without failure.

Another example was when our gear shift mechanism broke during a climb in the Pacific Northwest Championships, which

1 See details for FIRST Robotics Competition at www.firstinspires.org. More than 80% of the FRC teams are from the USA, but many other countries are represented, including China, Brazil, Canada, Mexico, Israel, Turkey, etc. There are also competitions with smaller robots for 8–12th grade students and lower-level competitions for middle school students which involve Lego-based robots built from robotics kits using simplified software. There are over 400,000 students involved in FIRST robotics at all levels around the world.

2 See a typical competition for our team at https://www.youtube.com/watch?v=uZlZbhmbNC4. Note that some defense is played which contributes to the wear and tear on the robots.



CAD drawing of the Robot Enterprise – Team 4450

had never happened to us before. The gears have a power take-off function for climbing and the gears had shifted so that they were wearing out under the hard use. We had to take apart the robot down to the level of the gears which took 3 hours and this was a major issue because we had to play a match without the ability to climb, and we just barely lost the match. That was a big penalty for us and a major learning point about design, accuracy of construction, failure points and maintainability. If we had made closer analysis of the location of the forces on the different parts of the climbing mechanism we could have made design trade-offs to reduce the potential for failure and increase overall robot reliability.

Another aspect of reliability analysis took place in the design of the autonomous program (written in Java) to allow the robot to place a gear on the airship. Initially we used a dead reckoning program that counted the rotations of the wheels to arrive at a key pivot spot and an approximation of the angle to take to the airship. This led to only about 50% success in placing the gear exactly on the target spring support. To improve this we integrated a camera with image recognition based on light from reflecting strips located above the support. The image location was then fed back to a program which adjusted the angle of the wheels as the robot approached the target. This increased our reliability to about 75%.

Our struggles with various aspects of reliability continue to raise questions about how the team should prepare for better design. This is a teenage team with very good pit repair skills and good CAD skills for robot design. They are not a team of professionals and do not have all the concepts they need or the ability to carry out a large number of tests in the six-week build season. There was also a lot of hand reworking of parts due to inaccuracies of the CAD design or last minute changes, which limited our practice and testing time. We are now assessing the type of preparation we can do for next year's challenge.

In addition to working with robots, our team reaches out to the community to support STEM education and to mentor lower level robotics teams. We also learn leadership, marketing communications and how to run a small business. This is a fascinating combination of learning, teaching and fun for the mentors as well as the students. Mentoring a team is a good way for professionals to give back to their communities and we encourage you to make contact with your local teams and offer your support. Local businesses can also offer donations of cash, materials (e.g., aluminum and Lexan) and services to help make our teams successful. Become a mentor find a team now![3] ●

3 See https://www.firstinspires.org/team-event-search#-type=teams&sort=name&programs=FRC&year=2017 for the location of FRC teams in your area and to identify upcoming events and competitions.

Robots climbing in competition

QUIZ YOURSELF

An electro migration model for testing ($MTTF = AJ^{-n} e^{Ea/KT}$)...

a) ...is a chemical oxidation-subduction reaction.
b) ...requires a layer of water approximately 5 monolayers thick for ECM to occur.
c) ...current density to a power and the Arrhenius temperature dependence.
d) ...associates ECM on printed circuit board assemblies (PCBSs) as conclusively demonstrated.

Prenscia HBM

Synthesis Platform Version 11 in now available

ReliaSoft Corporation continues its ongoing commitment to providing the leading solutions for reliability engineering, quality and maintenance planning needs of product manufacturers and equipment operators.

WEIBULL++ ALTA RGA BLOCKSIM RENO λPREDICT

XFMEA RCM++ RBI MPC XFRACAS SEP

The Synthesis Platform applications now offer new functionality and more value through integration. Here is a short list of some of the exciting developments you can take advantage of by upgrading to Version 11:

- Weibull++ and ALTA offer full Design of Experiments capabilities
- Xfmea, RCM++ and RBI now allow you to create Parameter Diagrams (P-Diagrams), which provide a visual method for documenting input signals, noise factors and control factors that lead to ideal and undesirable system responses
- Siemens SN 29500 reliability prediction standard is now available in Lambda Predict
- The Synthesis Enterprise Portal (SEP) has a fresh new look with responsive design for better performance on mobile devices to access analyses without having the Synthesis applications installed
- Improved performance and updated interface for spreadsheets
- ALTA Stress Profiles and nCode Glyphworks integration to analyze time series data (*.S3T files)

For more information, visit our website at http://www.reliasoft.com or email us at sales@reliasoft.com

Editorial, from page 1

matter consultation guidance as our primary tools in working with an increasingly diverse number of new industries.

This year we are planning to cast a wider net to include industries that are confronting reliability, maintainability, supportability, logistics and systems engineering challenges. We will begin to develop working relationships with the transportation, environmental, energy, construction and medical industries, just to mention some new fields of our interest.

Few of these industries have been as successful as the defense community in addressing life cycle system engineering and logistics issues. Also there are emerging specialties such as robotics, Internet of Things (IOT), connected cars (advanced driver assistance systems) and other areas that could benefit from the background and experience that the RMSP has accumulated primarily from its close working relationship with the defense sector. Conversely, by expanding its reach into other disciplines, the RMSP certainly will acquire information and knowledge that should prove to be mutually beneficial to the defense community.

The transformation of the RMSP to include broader professional occupational sectors within the national and international global economy requires new long term strategic and implementation planning. We will reach out to many professionals to provide us insight and support in identifying ways we can work together on resolving numerous technical and managerial challenges. Initial efforts will include providing more diverse newsletter and professional journal articles, hosting interdisciplinary-professional workshops and conferences, and expanding the RMSP membership and our readership. At the same time, we will continue to maintain rigorous training and related activities for the government-industry defense community.

All those who are interested in working with the RMSP on our new goals and objectives are asked to contact me at president@rmspartnership.org. Volunteers and those interested in earning "business development" commissions are also encouraged to contact me directly. Course developers and experienced instructor opportunities continue to exist. At present the RMSP offers 38 short courses that can be offered online or on-site. Please visit our website, www.rmspartnership.org, to read a short description of these courses.

In the coming year the RMSP will widen its professional outreach to include new industries that are interested in acquiring the knowledge and understanding to improve systems reliability and reducing total life cycle costs. The exchange of such information should also assist these industries in reducing potential systems venerability to internal and external threats.

With your support and cooperation we can continue working on these worthwhile activities. I look forward to hearing from you soon. ●

About the Authors

Ann Marie Neufelder has 34 years of experience in the field of software engineering and software reliability. She has invented a patent-pending method to predict software reliability prior to the code even being developed. Since 1993, she has benchmarked the software reliability of more than 150 organizations in the defense, aerospace, space, medical, energy, and chip industries. Ann Marie has also identified more than 400 software failure mode/root cause pairs as well as a process for performing software failure modes effects analysis. Ann Marie is the Chair of the 2016 re-write of the *IEEE Recommended Practices for Software Reliability*.

Peter Cook is an engineer, economist, and small-businessman who has been a community volunteer working as Lead Mentor with a high school robotics team for the last five years. His team has qualified for the World Championships in three of their five years, winning awards for excellence in engineering and innovation in control systems.

Before that he led a 35-year career in international transportation planning, logistics, and information systems. He worked on projects ranging from small transportation studies to national and regional-level transport planning and policy studies in over 60 countries. During this career he worked with small teams from several countries, including India and China, to help them create innovative analytic tools for regional and national investment and policy analysis. He is the author of numerous published papers and has taught university-level courses. He is a graduate of MIT and Harvard.

INTERESTED IN CONTRIBUTING?

If you are interested in sharing your knowledge in future editions, please contact Russ Vacante at: *president@rmspartnership.org*
Articles can range from one page to five pages and should be of general interest to our members.