Optimizing the Design of Spacecraft Systems using risk as currency[1]

Steven L. Cornford
MS 179-224
(818)354-1701
steven.cornford@jpl.nasa.gov

Julia Dunphy
MS 126/202
(818)393-5365
julia.dunphy@jpl.nasa.gov

Martin S. Feather
MS 125-233
(818)354-1194
Martin.S.Feather@jpl.nasa.gov

Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109

Abstract- Treating risk as a "currency" has proven to be key in systematically optimizing the design of spacecraft systems. This idea has been applied in the design of individual components of spacecraft systems, and in the end-to-end design of such systems. The process, called "Defect Detection and Prevention" (DDP), its tool support, and applications, are described in [1]. The process can be summarized as determining the consequences of various risk elements (technical, programmatic, other) on the various requirements and then determining the optimal combination of approaches to reducing the risks to acceptable levels.

We are now extending this process to include consideration of architectural alternatives, qualification of components, fabrication and assembly, integration and test, and mission operation. The results of applying this extended process in the pre-formulation, formulation and implementation phases of various NASA and other government agency missions will be discussed. This paper will also discuss the results of developing optimized technology development and qualification plans.

TABLE OF CONTENTS

1. INTRODUCTION

The formulation, implementation and operation of complex spacecraft systems is a challenging endeavor. There are usually significant amounts of advanced technology, the various subsystems are composed of state-of-the-art components and the integration of these subsystems into a spacecraft is a complex task in its own right. To further complicate the task of deploying complex spacecraft, there is significant pressure to reduce the schedule, cost and risk of these missions. Finally, many of these spacecraft rely on the parallel maturation of advanced technologies in order to achieve aggressive mission objectives.

Thus, one can imagine a risk landscape covering many dimensions including time (schedule), performance (or utility), risk level, risk type, and available resources. Once one can make risk a measurable quantity versus the other dimensions one could generate such a risk landscape surface. Many useful views would then be possible which would be of significant utility to decision makers:

Let us first examine the view show in **Figure** 1 which plots the risk versus schedule (time) for fixed performance, fixed resource constraints: From this view, one can see that while Option C promises the lowest operational risk it has deferred much of its life cycle risk until late

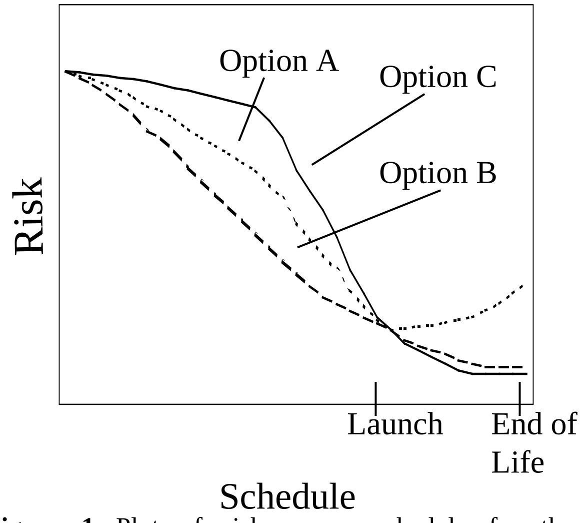


Figure 1 Plot of risk versus schedule for three hypothetical project options. Note the differences in deferred risk versus schedule for the three options.

[1] 0-7803-7231-X/01/$10.00/© 2002 IEEE

implementation, and thus, has a lower probability of achieving the operations advantage. Option B appears to be the best options since it addresses much of its risk very early and achieves lower operations risk as well. Explicit trades of early application of resources to "buy down" downstream risk are now possible. Note that the risks are reducing as we implement solutions (prototypes, simulations, etc.). Also note that the repair of a discovered risk element costs more money, takes more time and usually has more system 'ripple effects' the later in the life cycle it is addressed.

Let us now examine the view show in **Figure 2** which plots the risk versus schedule (time) for fixed schedule and resources for some performance parameter. From this view, one can see that instead of only Performance level A one can obtain Performance level B for only a small incremental increase in risk (which can be further reduced by additional/earlier resource allocations. Performance level C is still too risky and could benefit (presumably) from technology maturation or re-architecting. Note that the A, B and C used to differentiate performance levels, are not the same A, B and C used in the previous figure to describe project options.

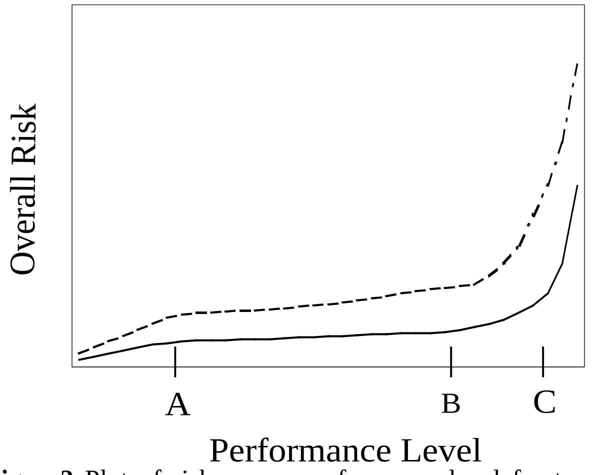


Figure2 Plot of risk versus performance level for two hypothetical projects. Note the large gain in performance level for only minimal increases in risk in the region from A to B, but the large increase in risk for minimal increases in risk in the region to the right of B.

As motivational as these views are, we must return to today's state of affairs to see how we can get there.

2. DDP Process Review

The DDP process has been discussed in detail in a previous publication [1] but is summarized briefly as: determine what we are trying to accomplish, determine what could get in the way and determine what we can do about these potential stumbling blocks. The process is also summarized graphically below in **Figure 3**. Note that trees of potential failure modes (or risk elements) are generated and their impacts (and likelihood) are evaluated against trees of weighted requirements in the R matrix. The E matrix then evaluates the effectiveness of various PACT (Preventative measures, Analyses, process Controls and Tests) options against the potential risks. The user can then select various PACTs to mitigate (prevent or detect) various risk elements and the tool keeps track of the cumulative resource costs for the selected PACT combination.

The engine computes using the entered information – the a priori likelihood of each risk element, the impact of each risk element upon each requirement, the effectiveness at detection/prevention of each PACT versus each risk element. This internal computation is fairly simple and performs weighted sums (across the rows of the R matrix) to generate the requirement impact, and weighted sums (down the columns of the R matrix) to generate the initial risk. The engine also computes the residual risk by evaluating the product of the effectiveness of the selected PACTs for each of risk element.

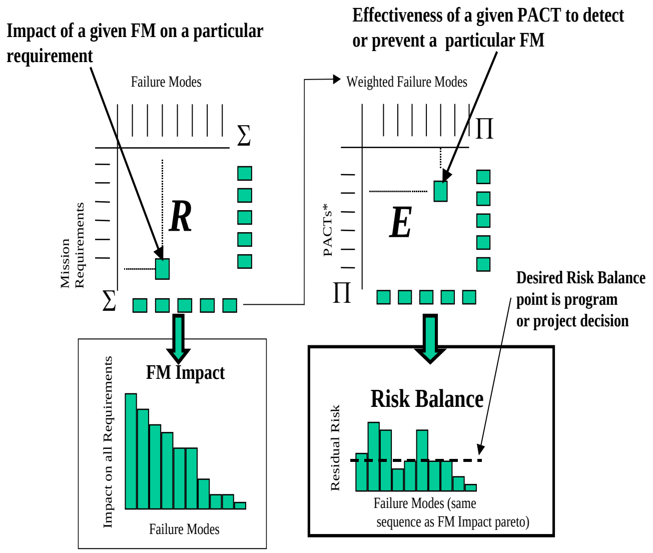

Impact of a given FM on a particular requirement
Effectiveness of a given PACT to detect or prevent a particular FM
Failure Modes
Weighted Failure Modes
Σ
Π
Mission Requirements
R
Σ
PACTs*
E
Π
Desired Risk Balance point is program or project decision
FM Impact
Impact on all Requirements
Failure Modes
Risk Balance
Residual Risk
Failure Modes (same sequence as FM Impact pareto)

Figure 3 Graphical summary of the DDP process. The Requirements matrix results in a prioritized set of Failure modes (sorted in the bar chart), a representation of the driving requirements (not shown), and the residual risk if the selected PACTs are implemented.

The DDP process is a near real-time risk management process in the sense that it uses all currently available information to generate a collection of risks which can then be reduced, re-ordered/balanced, refined or accepted by the application (or not) of various detection and prevention activities. Early in the project life cycle, much of this information is generated in real-time during half-day sessions involving a 'critical mass' of experts. However, existing information (previous analyses, test results, information of previous projects, etc.) is integrated as it is available and relevant.

Note that one can evaluate risks in more detail and with greater fidelity (e.g. simulations and prototypes) as the size or uncertainty in the risk warrants. Thus, the DDP process can be used as a front-end to a more detailed probabilistic calculation by identifying the tallest poles for additional evaluation. Ideally, the work a project is doing anyway would be addressing those areas where risk is greatest and/or most uncertain. However, as we shall see the project life cycle tends to result in project focusing on those risks which are immediately impacted rather than those lurking downstream.

3. Towards Risk as a Resource

Risk is usually thought of as bad luck or oversight in the sense that it is a consequence of the good work we have already done. Thus, risks are those items which escaped our carefully orchestrated processes. However, we submit that merely defining risk as the inability (still impact times likelihood) to meet requirements opens up a new way of considering risk over the entire life-cycle.

If risk is to be considered a resource one must define what we mean by risk in early design cycle phases. Keeping in mind that risk is the inability to meet requirements, we can use more generalized notions of risk to watch the risk landscape evolve over the life cycle.

Generalized Notions of Risk

Consider the following risk categories:

Nominal Function [N]: The inability of the design to perform its basic function(s).

System Compatibility [S]: The inability of the various system constituents to function as an integrated system.

Robustness/Resiliency [R]: The inability of the system (or constituents) to function normally in unexpected situations or environments.

Environmental Compatibility [E]: The inability of the system (or constituents) to function normally over the expected environments (ground, launch, vacuum, etc.).

Manufacturing and Assembly [M]: The inability of the system (or constituents) to be manufactured, assembled or integrated.

Parts and Materials [P]: The inability to obtain or utilize the desired parts and materials required for the desired end product.

Humans and Organizations [H]: The inability of humans or organizations to perform the functions necessary to produce the desired end product.

These categories of risk are for discussion purposes only and there may be a more complete, or logical grouping. However, using these categories we can examine the degree of attention which is focused on the issues within each category. In many cases, the early (formulation) portion of the project life cycle is focused on ensuring that the basic system requirements are achievable and self-consistent. Later in the implementation phase, more attention is focused on the Environmental Compatibility, Parts and Materials and Manufacturing and Assembly risk categories. Finally, as launch approaches, more attention is focused on the robustness and Human categories. We are not saying this is how everyone approaches the addressing of risks in these categories, but we are saying this is not the right approach.

This discussion is illustrated in **Figure 4** where the size of the gray boxes represent the degree to which attention is focused on the reduction of risks in these areas. Note that bars to the right are a representation of the residual risk in each area at various phases and are ordered to show a ‘bow wave’ of risk as areas are addressed with differing levels of attention in the project life cycle. This risk bow wave is result of a series of sub-optimizations in which we first determine the architecture and system design, then determine how to design it within the architectural and system constraints, then determine how to implement it within the design constraints, then determine how to make it robust given the implementation constraints, then figure out how to operate it given the robustness constraints. Given the increasing cost of fixing a problem as the design matures, it is obvious that there would be large payoff from focusing some more attention initially on the potential implementation, robustness and operational risks associated with a given architecture, or design.

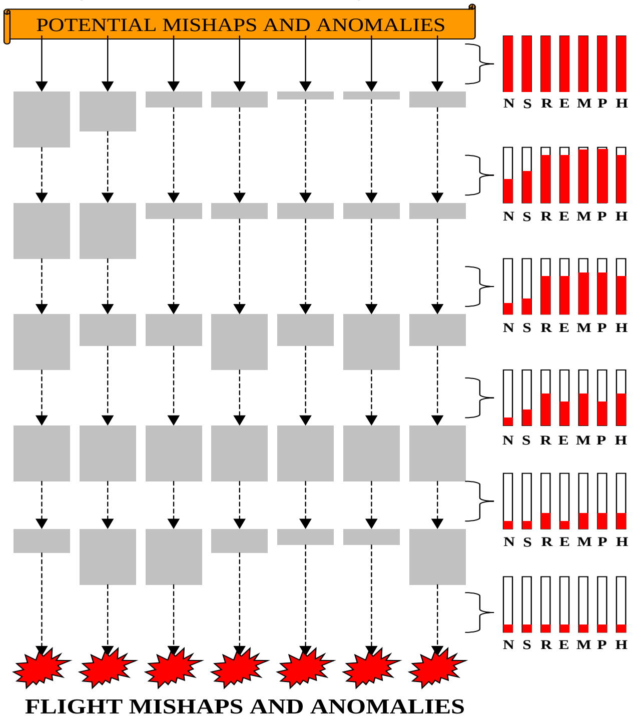


Figure 4 Chart illustrates the risk “bow wave” which results from optimizing the risk reduction in each project phase and not over the whole life cycle.

Generalized Notions of PACTs

PACTs for detecting or preventing risk can be thought of as more than tests or analyses. When a risk is the risk of having inadequate data processing speed, PACT options include various processor types (each of which may have additional risks associated with them). Note that the architecture and design of the system now entails selecting PACTs to minimize the risks identified at that phase.

4. Optimization of Risk Reduction Activities

The primary purpose of DDP is to enable users to emerge with a judicious selection of PACTs (risk mitigation activities). Each selected PACT has benefits – it reduces the risks against which it is effective (and thereby increases the attainment of the requirements impacted by those risks), but incurs costs – dollars, schedule, allocation of mass, electrical power, etc. Judicious selection means picking a set of PACTs that together ensure *cost-effective* attainment of requirements over the life-cycle.

To date, DDP has relied on users to manually select PACTs, using several cogent visualizations of the relevant information to assist them in this task. Recently, we have begun the automation of this selection process, by casting it as an *optimization* problem. For example, the users can set a cost cap, and pose the optimization problem of finding the most effective set of PACTs whose sum total cost does not exceed that cap.

The scope of the optimization problem depends on the user constraints. Ideally, one would maximize the expected value of the return (e.g. expected value of requirements achieved) with respect to cost. However, the realities of interplanetary flight development process results in a variety of additional constraints including cost and schedule as well as pre-determined science objective constraints. There may be a variety of other factors including availability of launch compatibility issues, and round-trip light travel time to name a few. Thus, the typical optimization problem encountered is to maximize the science return subject to the various constraints. Since the DDP process evaluates risk element significance by impact on requirements (times the likelihood of occurrence), maximizing the science (requirement) return is equivalent to minimizing the risks for the given project.

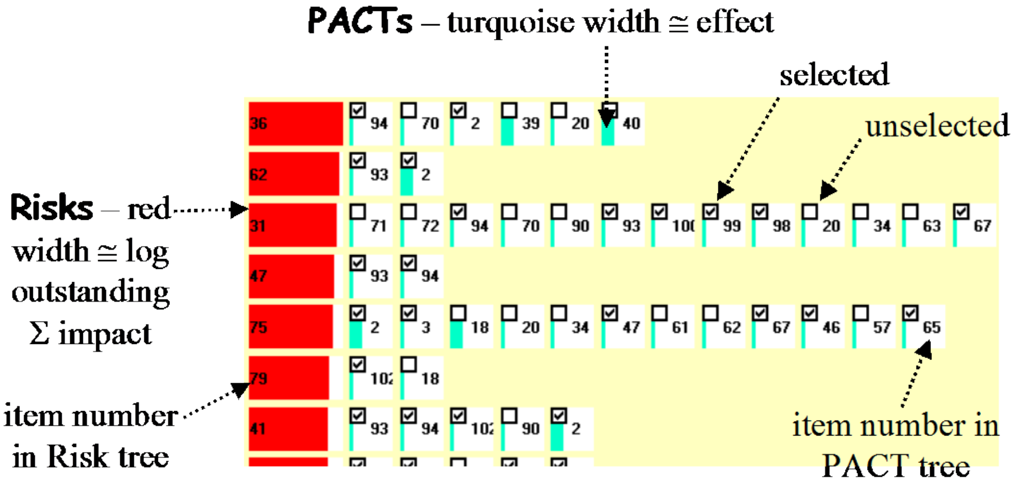
Application of the process has revealed an interesting degree-of-freedom in the optimization problem: the relative importance of the various requirements. Since requirements are weighted, changing their relative weights can have significant effects upon the risk balance…

The subsections that follow present DDP's support for manual selection of PACTs, for automated optimization, and finally a brief discussion of elaborations to the cost model.

Manual selection of PACTs

To date, DDP has relied on users to manually select PACTs. To facilitate this activity, DDP provides the following visualizations of information relevant to this task:

Risks x PACTs lists – this visualization presents a list of all the risks, and for each risk, a list of all the PACTs applicable to reducing that risk. A snapshot of a portion of such a view is shown below in **Figure 5**. This view allows the users to see for a given risk which PACTs mitigate that risk, and by how much. Furthermore, a PACT may mitigate multiple risks, in which case it appears in each of those risks' list of PACTs. Such occurrences are made visible by flashing all instances of a PACT when the user moves the mouse over any one of them. This is an interactive view – users can both see, and change, the selection of PACTs by clicking on the check box on a given PACT to toggle its 'selectedness'.


PACTs – turquoise width ≅ effect
selected
unselected
Risks – red width ≅ log outstanding Σ impact
item number in Risk tree
item number in PACT tree

36	☑ 94	□ 70	☑ 2	□ 39	□ 20	□ 40							
62	☑ 93	☑ 2											
31	□ 71	□ 72	☑ 94	□ 70	□ 90	☑ 93	☑ 10(	☑ 99	☑ 98	□ 20	□ 34	□ 63	☑ 67
47	☑ 93	☑ 94											
75	☑ 2	☑ 3	□ 18	□ 20	□ 34	☑ 47	□ 61	□ 62	☑ 67	☑ 46	□ 57	☑ 65	
79	☑ 10:	□ 18											
41	☑ 93	☑ 94	☑ 10:	□ 90	☑ 2								

Figure 5 Snapshot of the Risk x PACTs list view available in the tool. This view is a better way to view the relationships between matrix elements when the matrix is sparsely populated.

There is the analogous view of PACTs x Risks lists, in which each PACT is listed, and alongside each PACT, the list of all the risks is mitigates.

PACTs bar chart – this visualization presents a bar-chart where each bar corresponds to one PACT. The height of the bar is proportional to the sum total effectiveness of the PACT (i.e., the total amount of risk reduction it achieves). We have found it useful to calculate and display two variants of total effectiveness for each PACT: the “solo” effectiveness – the sum total risk reduction it would achieve if it were selected in isolation, and the “delta” effectiveness – the incremental risk reduction it would achieve if it were selected in addition to the other already selected PACTs. These two totals can be quite different, due to the fact that other already selected PACTs might significantly reduce a risk, so selecting an additional PACT has less of an incremental risk reduction than its solo effectiveness (this is a consequence of the way that effectiveness of multiple PACTs combine).

Risk bar chart coupled with PACT viewer – this visualization (shown in **Figure 6** below) presents a bar-chart where each bar corresponds to one risk. The height of the bar is proportional to the sum total impact of the risk (i.e., the total amount of requirements loss it causes).We calculate and display two impact totals for each risk – the sum total impact ignoring PACTs, and the sum total impact taking the mitigating effect of the currently selected PACTs into account.

The bars can be ordered in one of three ways:

1. by the occurrence of the risks in the Risk tree (as shown in the figure),
2. sorted once in descending order of remaining risk, and held in that order
3. sorted in descending order of remaining risk and continually resorted as this changes.

Finally, the user can suppress the view of the original risk level, and choose instead to see deltas (increases in risk and decreases in risk) with respect to a previous PACT selection set. The combination of these features allows the users to switch between views that show the progressive and/or comparative effect of PACT selections, and (notably by sorting) to see the outstanding risks remaining to be dealt with. When the user clicks on a risk’s bar, an accompanying window shows all the PACTs applicable to mitigating that risk, how much each PACT costs, how effective it is, etc. In manual selection of PACTs, we have found users to employ this pairing of windows to make most of their PACT selection decisions.

Resource meter – this visualization presents a running tally of the resource costs of the PACTs currently selected. Currently, resources used are mass, power and $, but additional user-defined database fields are available to allow other resources to be tracked.

Automated optimization

The selection of the best possible set of PACTs is a daunting one. A typical project may have as many as 100 or more possible PACTs. Cost restrictions limit the number of PACTs that can be chosen, but the number of combinations that have the same or similar cost is astronomical. For example, a blindly exhaustive search for the permutations of the 100 PACT options results in 2^{100} options which would require at least 300 years to evaluate (if we had 10^{13} Gflop computing capability!). Selecting among these combinations requires calculating how much of the original requirements would be preserved by each candidate set. This calculation itself is not trivial even ignoring interactions between PACTs. As can be seen, this kind of “selection” optimization is not amenable to some standard techniques such as those found in those based on continuous variables such as operations research. While there is some grouping of PACTs and their effectiveness versus risk elements (e.g. various optical alignment design and test options all have some effectiveness versus a specific subset of risk elements), in general, each candidate subset of PACTs is unrelated to any other. There is no obvious way

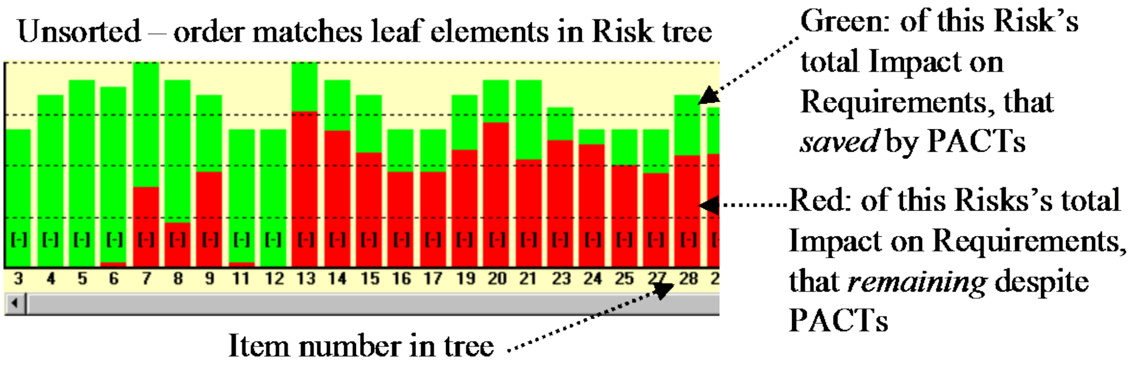


Figure 6 Snapshot of the risk bar chart view available in the tool. The numbers at the bottom of each bar correspond to and identified risk element.

to gradually transform the candidates by seeking local or global minima or maxima, but this is still being explored and will be discussed later in this paper.

One approach we have considered and implemented in the latest release of the DDP software tool, is a genetic algorithm based on some work done at JPL on designing quantum circuits. A quantum circuit is a set of quantum logic elements whose overall transition matrix matches the desired behavior. This problem has many of the same characteristics as ours. The number and variety of logic elements is relatively small but each circuit is independent of all others. The genetic algorithm attempts to make small changes to identical copies of quantum circuits and test their nearness to the desired solution. The nearness to the desired behavior is used to bias the next generation to employ the elements that appear to be helping to achieve the goals. Thus, it is an 'evolutionary' process with the 'best' candidates breeding the most candidates for the next iteration. The "small changes" include moving the logic element to a different wire, changing the control element, adding or removing a random element and similar. In our case, it is rather harder to define small changes, but we can add or remove a single PACT from the candidate set . We have implemented this algorithm as a new feature in DDP version 3.0. It should be emphasized that this optimization technique is not guaranteed to produce the 'best' solution (i.e. global optimum), but testing to date on tractable problems has found excellent agreement. We are still exploring mutation parameter sensitivities.

Another approach under investigation is due to Tim Menzies who is on contract to JPL. He has developed an approach where many sets of PACTs (candidate solutions or "treatments") are run and the members that seem to have a very large effect, either positive or negative are identified. Tim has shown that there are always a few members of a treatment that have a much larger effect than others and these can be used as a basis for the best treatment. These "outriders" are PACTS that always seem to be identified with treatments that produce large percentages of surviving requirements while maintaining cost constraints. We are hoping that these members can be forced into new treatments and other lesser, but still important PACTs identified in a recursive way.

We are still exploring other methods of optimization. We may be able to take advantage of the fact that most PACTs can only be performed at a single phase of the project. By "phase" we mean preliminary design, breadboarding, prototyping, final build, system test etc. Suppose we have the typical five or six phases each with about an equal number of PACTs. Then we only have to consider searching for the best 5 out of 20 for each phase and that is a far more tractable problem. It turns out that NASA and many other funding sources schedule the money to perform these PACTs by phase so the amount of money available in each phase is usually known ahead of time. This fact leads to a very simple fit of the real funding profile to the tractability of the exhaustive search – a fortunate co-incidence. If we optimize by phase, we do introduce the possibility that we are missing an overall global solution that might achieve the same degree of requirement survival at a lower overall project cost. However, political realities do not usually allow us to divide up the project budget into the project phases any way, but cause us to follow predetermined funding profiles.

In our implementation of the optimization process, we believe that we should show to the operator all the intermediate results, since this may give the operator further insight about the optimization process. In the case of the genetic algorithm, we show all the considered solutions and their score in terms of the percentage of the requirements that are preserved by them. It is possible that a user might notice a pattern that it not easily discernable to the code that the user could take advantage of.

Elaborated cost model

In recent months we have elaborated the DDP cost model Full details of this model are given in [2]. Briefly, the elaborations are to:

- Make a distinction between different categories of mitigations - preventions, detections and alleviations.
- Separate the cost of performing a detection-style risk mitigation from the cost of repairing the problems it detects.
- Assign detection mitigations to distinct phases of development, permitting the calculation of repair cost to take into account the phase in which the repair is conducted.

These elaborations have significant consequences for PACT selection, whether done manually or automatically. In particular, they extend the intertwinedness of PACTs. For example, in this elaborated cost model, it is possible to select an additional PACT and see both the benefit increase and the total costs decrease! This occurs when the PACT discovers (or prevents) problems early on that would otherwise be discovered and repaired in later stages, by which time their repair costs are significantly higher. Thus, we can now explicitly introduce the well-known (but poorly documented) 1:10:100 rule, where it costs 10X more to fix a problem at each level of integration.[2]

The net result is that an incremental selection strategy, in which users manually pick PACTs one by one, is less viable as a means to arrive at a satisfactory selected final set when managing risks for large-scale projects.

[2] One can image various designs which would follow the $1:x:x^2$ rule, where x can be larger, or smaller, than 10. For example, highly modular designs should result in x less than 10.

5. Summary and Conclusions

The DDP process has been described and provides an illustration of the utility of considering more generalized notions of risk. This results in a risk landscape in which risk is the currency by which one navigates the landscape – one can reduce risk by applying more PACTs or selecting a different combination of PACTs. The consideration of life-cycle risk has been shown to be important in reducing the pain associated with the late implementation phase of nearly all space-flight development projects. It has been shown that earlier application of resources may not only result in lower risk, it may also result in lower cost!

This leads to the question of how to optimize the PACT selection over the entire project life cycle. Several approaches to solving the optimization problem have been presented and some preliminary results have been discussed.

Future work includes deploying additional optimization capability and developing optimization approaches for the case where the impacts, likelihoods and effectiveness values have additional uncertainty associated with them.

Acknowledgements

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

References

[1] Steven L. Cornford, Martin S. Feather, Kenneth A. Hicks, "DDP – A Tool for Life-Cycle Risk Management", *2001 IEEE Aerospace Conference Proceedings*, Big Sky, Montana, January 2001

[2] Martin S. Feather, Steven L. Cornford, Julia Dunphy, "Cost-Benefit Based Assurance Planning", *International Workshop on Model Based Requirements Engineering Proceedings*, 2001.

Biographies

Steven Cornford *is a Senior Engineer in the Strategic Systems Technology Program Office at NASA's Jet Propulsion Laboratory. He graduated from UC Berkeley with undergraduate degrees in Mathematics and Physics and received his doctorate in Physics from Texas A&M University in 1992. Since coming to JPL he focused his early efforts at JPL on establishing a quantitative basis for environmental test program selection and implementation. As Payload Reliability Assurance Program Element Manager, this evolved into establishing a quantitative basis for evaluating the effectiveness of overall reliability and test programs as well as performing residual risk assessments of new technologies. This has resulted in the Defect Detection and Prevention (DDP) process is the motivation for this paper. He received the NASA Exceptional Service Medal in 1997 for his efforts to date. He has been an instrument system engineer, a test-bed Cognizant Engineer and is currently involved with improving JPL's technology infusion processes as well as the Principal Investigator for the development and implementation of the DDP software tool.*

Martin Feather *is a Principal in the Software Quality Assurance group of NASA's Jet propulsion Laboratory. He works on developing research ideas and maturing them into practice, with particular interests in the areas of software validation (analysis, text automation, V&V techniques) and of early phase requirements engineering and risk management. He obtained his BA and MA degrees in mathematics and computer science from Cambridge University, England, and his PhD degree in artificial intelligence from the University of Edinburgh, Scotland. Prior to joining JPL, Dr. Feather worked on NSF and DARPA funded research while at the University of Southern California's Information Sciences Institute. For further details, see* [*http://eis.jpl.nasa.gov/~mfeather*](http://eis.jpl.nasa.gov/~mfeather)

Julia Dunphy received her Master's and Bachelor's degrees in Physics and Mathematics from Cambridge University, UK, ('63) and her doctorate in Theoretical Physics from Stanford in '67. After a career in magnetic recording research in the 70s, she switched to software development and was a cofounder of a small company which provided development software for the then infant microcomputing industry. She now works as a contractor to JPL in the areas of design research and network computing. Her interests include the field of collaborative engineering design infrastructures and automatic source code generation. She holds several patents and has published over two dozen papers in various areas, such as magnetic recording, error-correction coding, and control of robotic vehicles (for the Mars Pathfinder Rover).