Learning from Failure:

Managing Changing Requirements on the Apollo 13 Mission

***Abstract.*** In every engineering project there are a multitude of requirements to manage. These requirements are crucial to the successful completion of the project, but if not properly managed they can have negative consequences and even potentially disastrous results. This paper examines how even the smallest change in requirements, not properly disseminated to all the stakeholders led to the failure of the famous Apollo 13 mission and almost cost the lives of the three astronauts on board. This paper will also examine some of the techniques which can be used to manage changing requirements as well as avoid the problems of creeping requirements.

1. **Introduction**

“Houston, we’ve had a problem here...” are the words that astronaut Jack Swigert spoke after the crew of Apollo 13 heard what was described as a loud bang. Soon after the noise the command module’s caution and warning lights began illuminating. Mission control and the crew began seeing many oddball readings from the spacecraft. In fact, an instrumentation problem was initially suspected by engineering. However soon all hope of instrumentation failure was lost when astronaut Jim Lovell radioed that something was venting out into space. It turned out that it was liquid oxygen venting into space from damaged service module.

 The “loud bang” heard by the crew was actually oxygen tank number two exploding inside the service module of the Apollo 13 space vehicle. This occurred when the fans were turned on to stir the tank, but the problem was the result of a requirements change that happened years earlier. In 1965, three years after the contract was awarded to construct the command and service module, NASA decided that it wanted the then 28 volt electrical system to be completely compatible with the 65 volts system used in the ground-support and test equipment at Cape Canaveral in Florida [1]. North American Aviation at the time was awarded the contract to construct the command and service module, which would carry the astronauts to the moon and back (see picture 01 below). They subcontracted the design and construction of the liquid oxygen (LOX) tanks to Beech Aircraft. The thermostatic safety switches used inside the tanks were then subcontracted out to another supplier and were specified for 28-volts. When NASA requested the upgrade of the electrical system Beech was informed, but never passed the requirements to their subcontractor for the thermostatic switch, which remained rated at 28 volts [1].



**PICTURE 01: APOLLO 13 SPACE VEHICLE CONFIGURATION**

 This may never have been a problem because the heaters were rarely used, but due to an unrelated problem during the Apollo 10 mission the tanks were removed and transferred to the Apollo 13 ship. During the transfer tank number 2 was dropped about 5 centimeters damaging a drain line. A year later, two weeks prior to the Apollo 13 mission, the tanks were filled with liquid oxygen as part of a countdown test. Following the test the LOX tanks were drained, but the damaged line prevented LOX tank number 2 from draining so the ground crew activated the tank’s heaters in order to boil off the liquid oxygen. When the 65 volt ground power system was applied to the system, the thermostatic switch rated for 28 volts overheated and fused shut [2]. The purpose of the switch was to keep the temperature below 27°C, but with the switch fused shut the heaters remained on for eight hours as temperatures rose the Teflon insulation melted on the fan motor cables exposing the wires [2]. An additional discordance in requirements led to the thermometer associated with the system being calibrated to only 29°C, so no one even noticed that the overheating of the tanks was taking place. During the initial phases of the Apollo 13 mission the liquid oxygen in the tank prevented the wires from arching, but once the tank dropped to a point where the wires were exposed the activation of the fan caused the wires to arch and the tank to explode.

 The problem with the LOX tanks on Apollo 13 took place many years prior to the mission and was due to mismanagement of changing requirements. This event may have had disastrous results, but thanks to the training of everyone involved, the entire crew safely returned to Earth. This example illustrates the importance of proper management of changing requirements. The additional problem of the thermometer’s calibration shows why it is important to detect discordance among stakeholder’s requirements. The remainder of this paper will explore why requirements change and how to quantify the problems of requirements creep. It will also review some of the techniques used to effectively manage changing requirements by reducing the rate at which requirements change or making them less disruptive. Then we will look at how to understand and detect requirements discordances among stakeholders. Finally, we can use these techniques to understand why the requirements changed on Apollo 13 and examine how a different approach could have been used to ensure that changes were managed properly. Only through the proper understanding of past requirements issues can we learn to better control and manage changing requirements in order to avoid potential disaster on existing and future engineering projects.

1. **Why Requirements Change**

 As clearly illustrated by the Apollo 13 project, there will always be new requirements to deal with whether new government regulations come about, business priorities change, and even new stakeholders are identified. The need to change requirements during the course of any project is inevitable. Teams should anticipate changes and be adequately prepared to accommodate them. The following guidelines, prepared by IBM Corporation, will help teams be prepared to deal with changing requirements [3]:

* *Expect and plan for requirements that change throughout the develop­ment process.*
* *Continually reprioritize requirements based on changing circumstances such as business needs, customer importance, estimated effort and cost.*
* *Have a plan that you adjust at regular intervals.*
* *Keep your stakeholders informed as changes occur—get their input for prioriti­zation and the rationale behind it.*

While it is important to plan for change and continually evaluate existing requirements in order to adjust our plan, it is also necessary to understand why requirements change in the first place. One of the main reasons that requirements change is likely because we failed to capture all of them in the first place. Clearly the Apollo 13 example illustrates this principle by failing to capture the requirement to utilize the 65 volt ground power prior to launch at the beginning of development.

Capturing requirements implies that we understand or comprehend the requirements, record them properly and then communicate them effectively. While this process appears quite simple it is not as easy as it seems. For instance proper comprehension of requirements certainly depends on the level of knowledge each person involved has on the specific subject being discussed. Additionally most requirements may also involve some level of tacit knowledge transfer, which can only further complicate the process. The ability for a requirements engineer to comprehend what is tacit knowledge to a stakeholder and then convert that tacit knowledge to explicit knowledge in order to record it for effective communication to the project team is no easy task. One method for converting tacit knowledge to explicit knowledge and vice versa is known as the “spiral of knowledge creation” [4]. This method involves four steps which are socialization, externalization, combination, and internalization (see figure 01). Socialization involves creating analogies, metaphors or models to ensure that the tacit knowledge is understood properly from the stakeholder to receiver. Next the receiver must externalize the newly gained tacit knowledge by transferring it to an explicit knowledge that can be communicated with the project team. This explicit knowledge is then combined into the product or project that is being designed or built. The final step is to contribute to further expanding knowledge creation in the organization by sharing or internalizing what was learned across other projects. This final process converts the explicit knowledge into new tacit knowledge which the organization can benefit from as a whole [4].



**FIGURE 01**

 Another factor causing requirements to change is the dynamic business market environment. Rapidly changing economic, political, or social conditions can lead to the necessity to continually change requirements. Additionally rapidly evolving technology can significantly complicate the need for changes in requirements. In fact, a study by Edberg and Olfman in 2001 looked at the motivations behind software change requests. They concluded that corrective maintenance only accounted for 10-15% of work while functional enhancements accounted for over 60% of changes [5]. In order to maintain competitive advantage in today’s dynamic business markets it is important to be able to change rapidly. However, too much change can also cause problems. The problems associated with requirements creep will be explored in the next section.

1. **Avoiding Requirements Creep**

 As we discussed previously changes to existing requirements are inevitable and must be maintained to some degree in order to adapt to increased system needs. While change is necessary and must occur it is important to find ways to minimize the effects of changing requirements. There are two main ways of controlling requirements creep. They are to reduce the rate at which change occurs or make the changes less disruptive [6].

 In order to reduce the rate at which change occurs we must first understand how quickly change is taking place. One method of measuring rate of change is by using function point metrics. Function points are units of measure used for quantifying the functional requirements of a software deliverable based upon the user’s point of view. In order to utilize these techniques one must first size the application once the requirements are firm. The number of function points would then be compared to number when the development is considered complete. This information can be used to provide a direct measurement of the volume of creep [6]. To discover the rate of change one might analyze the evolution of requirements during varying phases of the project (design, development, etc.) creating a monthly rate of change. These changes are normally expressed as a percentage change to the functional point total from the original requirements specification [6]. Another way to express changing requirements is by looking at the average volume of change between the original requirements and the delivered application in terms of typical growth patterns derived from function point totals [6]. By measuring the rate of requirements change we can use this information to evaluate processes and procedures that will help reduce the rate of changing requirements or minimize their impact. However, if many changes are taking place we need to also evaluate the process for requirements elicitation to ensure we are properly understanding the requirements initially and eliminating discordances in requirements interpretation and evaluation.

1. **Detecting Requirements Discordances**

 In many cases changing requirements can be avoided by identifying discordances between multiple stakeholders’ requirements during the elicitation process. Requirements discordances are conflicts of interests and misunderstandings that can occur among stakeholders with regards to a particular requirement [7]. There are three possible problems which can occur when multiple stakeholders’ views differ on the same requirement. The first problem is a missing requirement, which occurs when stakeholder A has a requirement that stakeholder B does not possess. An inconsistent requirement is when both stakeholders have the same requirement, but different outcomes are expected or produced. Lastly a discordant requirement can be expressed in two cases as discordance in interpretation and in evaluation. This type of discordance is normally the result of a knowledge gap between stakeholders. In this case one engineer may interpret a requirement differently than the other based on their experience or understanding of the subject matter. This can in turn lead to varying degrees of evaluation of what is required to satisfy the same requirement and how important it is. In the Apollo 13 example this type of discordance may have occurred during the elicitation of the voltage requirements. Perhaps when the voltage requirements were discussed the space vehicle engineers interpreted the requirement as needing to use the standard 24-28 volts used by most aircraft, while the launch site engineers interpreted the requirement to mean the 65 volts that they are used to having on their ground power units. Another example is the thermometer used to evaluate the LOX tank thermostatic switches, which was only calibrated to 29°C. The interpretation of this requirement to measure the temperature of the tank and how it was evaluated by the different stakeholders led to obvious confusion about what was really required. In this case the interpretation of the requirement should have led to a thermometer requirement which would have satisfied the need to see an occurrence of over temperature on the system and not one that simply monitored operating temperature.

 Now that we understand discordance in requirements we need to be able to detect them so we can correct them in the elicitation process. One method of identifying discordances among stakeholders is through the use of attributed goal oriented requirements analysis (AGORA). This method of requirements analysis generates a goal graph and then creates preference matrices so that each stakeholder can assign values to their goals as well as those of the other stakeholders. Figure 02 provides an example of an AGORA goal graph with preference matrices. This allows analysts to easily detect interpretation and evaluation problems with requirements by reviewing the variance between each column of the matrix [7].



**FIGURE 02**

1. **Managing Changing Requirements**

 Now that we can measure and quantify the changes in requirements as well as prevent improper interpretation of requirements, we need to examine how we can control these changes in order to either reduce the rate at which requirements change or make the changes less disruptive. The following are some proven methods for helping reduce the rate at which requirements need to change or lessen the impact of these changes:

* 1. **Joint Application Design (JAD).** This method for developing requirements combines users and developers through the use of a facilitator in order to jointly develop requirements specifications. This process has been known to reduce creep rates by 50% [6].
	2. **Prototypes.** Building prototypes allow changes to move in front of the development cycle by allowing users to interface with the product. They can decrease creep rates by 10 to 25% [6].
	3. **Rapid Application Development (RAD).** This technique lends itself to applications with less than 1000 functional points. The idea is that the faster the product is developed the less time there for changing requirements. This is known to reduce changes by approximately 10% [6].
	4. **Requirements Inspection.** By applying formal inspection procedures to specifications and requirements errors can be detected and inconsistencies can be reduced. This method can reduce changes by almost 30% [6].
	5. **Cost-Per-Function-Point Contracts.** A reduction in changing requirements can be found by using these types of contracts which charge on a sliding scale; therefore the cost per function point rises the later the change occurs in the development process [6].
	6. **Quality Function Deployment (QFD).** In this method requirements are analyzed similar to the JAD method, but from the standpoint of user quality requirements. This method is typically employed in hardware development [6].
	7. **Change Control Boards.** This method is generally used for large systems in excess of 10,000 function points. It consists of a board made up of various stakeholders include clients, managers, and technical personal who decide which changes to accept or reject. During development of large systems this method can reduce the amount of changes by up to 25% [6].
	8. **Change and Configuration Management Systems.** Various commercial products are available to help control the speed at which changes can be processed as well as make it easier to ensure changes are properly controlled. These products will generally assist users with traceability as well as facilitating updates to a variety of documentation. Automation of change control can reduce effort by more than 70% [6].

 Since changes will occur throughout any development phase controls need to be in place to ensure that change is managed properly and accurately. However the amount of change management required can be significantly reduced by ensuring accurate and understood requirements are captured during the elicitation phase. This is why requirements elicitation phase of any development is the most important parts in the process.

1. **Application to Apollo 13**

 During the construction of the service module the requirements change requesting the use of 65 volts for the electrical system wasn’t passed down to all the stakeholders involved. There are also questions about why there wasn’t accurate traceability for this change? Also was there insufficient testing to validate the change in requirements? One could argue that a better process for managing changing requirements was the reason for the failure, but perhaps less discordances and a better process for requirements elicitation and evaluation was the reason the requirement wasn’t correct from the beginning. Regardless of the reasoning changes to requirements are unavoidable and therefore considerable effort is needed to ensure the proper processes and procedures are in place to manage these changes. Many of the techniques described in this paper can be employed to measure, monitor, and manage the process of changing requirements, but as illustrated by the events on the Apollo 13 no requirement is too small to overlook.

 The importance of traceability and testing as well as communication to all stakeholders must be stressed regarding all requirements. Furthermore, the process of eliciting the proper requirements in the beginning, although never perfect, should be one of the most important phases in the project in order to minimize changes requested during development. Once initial requirements are finalized, procedures to measure, monitor, confirm traceability, and verify through testing become critical to project success. While the failure of the Apollo 13 mission turned out to be a great success thanks to the tireless efforts of the engineers in mission control, it may have resulted in disaster all because of a small switch with the all the wrong requirements.

References

[1] S. Cass, "**Apollo 13, We Have a Solution**," *IEEE Spectrum,* 2005.

[2] M. Williamson, "Aiming for the Moon: The engineering challenge of Apollo," *Engineering Science and Education Journal,* vol. 11, pp. 164, 2002.

[3] IBM Corporation, "Getting requirements right: Avoiding the top 10 traps." 2009.

[4] I. Nonaka, "The Knowledge-Creating Company," *HBR,* July 2007.

[5] A. Kelly, "Why Do Requirements Change?" 2004.

[6] C. Jones, "Strategies for managing requirements creep," *Computer,* pp. 92, June 1996.

[7] H. Kaiya, D. Shinbara, J. Kawano and M. Saeki, "Improving the detection of requirements discordances among stakeholders," *Requirements Engineering,* pp. 289-303, October 2004.