Spreadsheets in Team X: Preserving Order in an Inherently Chaotic Environment

Keith Warfield
California Institute of Technology/
Jet Propulsion Laboratory
keith.warfield@jpl.nasa.gov

Jairus Hihn
California Institute of Technology/
Jet Propulsion Laboratory
jairus.hihn@jpl.nasa.gov

Abstract

JPL is NASA's prime center for deep space missions. In response to the need to reduce the cost and time to complete early concept studies and proposals JPL created the first concurrent engineering team in the aerospace industry: Team X. Started in 1995, Team X has carried out over 800 studies, dramatically reducing the time and cost involved, and has been the model for other concurrent engineering teams both within NASA and throughout the larger aerospace community. Since its inception, the software backbone of this highly successful design team – engaged in examining some of NASA's cutting edge concepts – has been the unassuming spreadsheet. Over the years the Team X spreadsheet-based tools have evolved from simple standalone engineering models into a networked spreadsheet intensive system with real time parameter updating. Recent new capabilities include stochastic cost estimation and a graphical drag and drop block diagram that automatically populates the related spreadsheet parameters of cost, mass and power. This paper describes how the spreadsheet functions within Team X: its history, architecture, current capabilities, enabling strengths and persistent weaknesses. In addition, the verification methods and institutional oversight that have evolved as the Team X products became increasingly critical to Laboratory success are also discussed.

1. Introduction

The Jet Propulsion Laboratory[1] (JPL) is a Federally Funded Research and Development Center managed by the California Institute of Technology for the National Aeronautics and Space Administration (NASA). JPL currently has 19 spacecraft and seven science instruments conducting active missions. The process of developing these missions is both expensive and time consuming as requirements need to be defined and competing architectures need to be fleshed out and compared to maximize science return for dollars spent. Typically, in the past, only a handful of studies could be examined due to budget constraints and these studies often spent years in the pre-project phase. Study teams assumed the role of advocate for the concepts that they were evaluating, losing objectivity and becoming "married" to design ideas that they personally had invested significant time and energy exploring but were many times inherently flawed or inferior to other ideas.

External events provided the motivation to challenge this problematic approach. To address the growing national debt, Congress began enacting a program of fiscal restraint that brought significant reductions in NASA funding in the mid 1990's. These reductions were taken as a challenge by the new Administrator, Dan Goldin who, in response, began a new movement within the Agency known as "Faster, Better, Cheaper" [1] wherein it was assumed that if mission cost could be made low enough and mission development be made fast enough, then some small risk of mission failure could be tolerated. The status quo was gone; efficiency and speed were the new paradigm. This movement was more revolutionary than evolutionary, and while it had its own flaws that eventually brought NASA back to a focus on mission success, it did (at least for a while) provide fertile ground for new, radical ideas that promised to improve the design process. So in 1995 – in an effort to increase the number of ideas examined, reduce the time and cost required, and return objectivity to trade space explorations – the Laboratory began an experiment in concurrent engineering identified as Team X.

[1] The research described in this presentation was carried out at 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.

Studies prior to Team X were carried out through weekly status reviews and action items; engineers, in large part, worked on their own between these weekly tag-up meetings. When consultations with other engineers were needed, they were carried out through phone calls (or often phone messages) and email. The shortcomings with this approach are many: communication is slow and much of it occurs at the weekly tag-up meeting; miscommunication within segments of the team is not uncommon and can go uncorrected for days; design information is captured in reports and presentations which are slow in creation and updated infrequently; trades are often narrowly focused and can omit key expertise that later alters or invalidates the earlier conclusion; and, as previously stated, study teams can develop a sense of personal ownership of a single design architecture due to a large personal investment in that option leading to struggles to "work out" flawed designs rather than move on to a better architecture.

Concurrent engineering promised to change all this. The idea was simple: bring together a team of engineers representing all aspects of space mission design and have them work out the design issues together. Meetings (referred to as "sessions" in the Team X lexicon) were for real-time design, not status review. Accordingly, scoping tools – mass, power, data, and cost estimation algorithms – needed to be available during the sessions. Many of the initial Team X engineers already had sizing tools in spreadsheets in their office computers so it was a short step to bring them all together in one room with a computer for each team member.

Hence, from the beginning spreadsheet based models were the primary software tools for the Team X concurrent engineering experiment and that relationship remains to this day. The Team X spreadsheet environment has evolved from a set of stand alone ad-hoc algorithms, to a system of distributed, highly integrated, organizationally endorsed models that interface with commercial software and are supported by an internal database. Today, the team's work can range from simple mass or cost estimation of existing designs to more complex optimization trades (e.g., trading trajectory vs. probe communications data rate vs. telecom energy requirements vs. battery mass vs. science return), to extended architecture studies (is a rover the most cost effective way to gather samples or would a hopper or group of low-cost landers produce better science?). And the resulting software environment that supports this capability is the largest most complex spreadsheet intensive system in NASA.

In the remainder of the paper, how the spreadsheets function within Team X – their evolution, architecture, current capabilities, enabling strengths and persistent weaknesses – are explored. Additionally, descriptions of the verification methods and institutional oversight that have evolved as the Team X products became increasingly critical to laboratory success are discussed.

2. Team X Spreadsheet Architecture

To understand how spreadsheets operate within Team X, some background on evolution of the Team X architecture and environment is required. The basic team currently consists of about 20 subsystem engineers (called "Chairs") with a Systems Engineer who oversees the data transfer mechanics of the session, and a Study Lead who orchestrates the study. Each Chair is responsible for the design, scope and cost estimate, and report write up for a particular subsystem. All have access to a Team X file server and the broader internet. The studies are loosely scripted and follow a general pattern of group briefing on assumptions, team design work, then reporting out on resource estimates. To the uninitiated, the sessions – specifically the design portion – can be a bit surprising since there is no effort to centralize or control the discussion and the room breaks down into many parallel conversations. People move about and the room can get loud giving the session a chaotic appearance, somewhat like the stock exchange. But like the stock exchange, there is an underlying order in the environment based on members' knowledge of the process and effective data transfer.

The team's now well established roles and processes have evolved over the years into its current, familiar pattern; likewise the team's use of the spreadsheet has also changed over the years. At the very beginning there were no sheets. The initial leadership of the team originally envisioned more sophisticated programs – mechanical and thermal finite element analysis, trajectory analysis, and the like – in part because there was a general disdain for "spreadsheet engineering" in the broader laboratory engineering community and the tools used in flight project development were viewed as more accurate and meaningful. But these tools can require significant design detail information (which is not available in the pre-project phase) and time to set up (which defeats a primary goal of concurrent engineering). The details of the team's studies vary considerably from one to the next so the day-to-day need is for a flexible tool that can be modified quickly and can support simple low-information,

low-resolution models; most of these high-resolution tools were largely shelved for special situations. In addition, early customers came in looking for simple, quick trade studies primarily involving mass and cost, and this remains a good part of the team's work to this day. These primary products of the concurrent study – mass, power and cost estimates – were ideally handled with spreadsheets since they were aggregates of individual subsystem estimates, so the initial need for spreadsheets was established with the first encounter with the team's customer base. Lastly, as stated earlier, many initial team members had already built spreadsheet-based sizing tools for their own personal use for pre-project subsystem scoping and trades so when asked to do the same in a concurrent environment they quickly brought in these tools for Team X use. Its ubiquitous nature, close functional alignment with the team's primary products, its simplicity, and adaptability quickly transformed what was envisioned at first as an ancillary support tool to the software backbone of the team.

The early team's spreadsheets were very ad-hoc. There was no attempt at configuration management; Chair members were free to change their tools as they saw fit. There was no real support or backing from the organizations that were responsible for the development of the subsystems modeled in the spreadsheets. (While not initially a significant problem, this lack of support from the line organizations would later become a major issue as the free-wheeling "Faster, Better, Cheaper" era came to a close resulting in an increase in institutional authority of the line organizations, and the Laboratory's need for convergence between the competing cost estimates coming out of Team X and the tasked organizations.) Still the increase in efficiency over dedicated study teams was undeniable, and the team's work increased as Program Offices saw the inherent value in concurrent engineering and brought increasing amounts of work to Team X.

Over the coming years this initial set-up underwent a series of changes that brought it to its current state. The first change was to link the sheets in a peer-to-peer arrangement. Difficulties in maintenance and scalability led to linking all the spreadsheets through a common data repository. From the beginning the team used this common database in a star configuration (all chairs interfacing only with the database – not each other) because direct interfaces between chairs grow geometrically with the number of chairs and maintenance quickly becomes unmanageable. Initially this database was simply a flat file contained in its own spreadsheet. The study's Systems Engineer was responsible for its set-up and operation during the study. Reading data from, and writing data to the spreadsheet were enabled through new functions within new versions of the spreadsheet software.

Connectivity quickly made the need for a common starting point apparent and a standard set of spreadsheet templates were developed for each of the subsystem chairs. The Systems Engineer would set up the study by copying the set of sheets to a new study folder and rename them to identify the study and conform to the standardized naming convention.

Success brought more complex architectures to study requiring more team members and more data transfer between the database and the Chair sheets. Soon a dedicated computer was added to support only the database file, so that the Systems Engineer would not be impacted by the team's read/write activities. The organizations responsible for designing and building the subsystems modeled in Team X began to participate in the Team X environment by taking "ownership" of their respective chairs (although ownership of the tools and Team X products were slower in coming) and became responsible for staffing them.

Finally, the flat file was replaced by a relational database. Many non-spreadsheet tools improved their interfaces to spreadsheet products and these tools began finding their way into the Chair sheets. And finally, the Lab began requiring the technical divisions to review their Chair's tools and endorse their Chair's products. Chair architectures are now being captured and reviewed with the owning organization's management and spreadsheet-based algorithms have undergone a year-long vetting process. This year, the Lab began requiring the use of Team X for all initial competed proposal cost estimates. Clearly what started as an experiment has become institutionalized and it still works largely with spreadsheets as it had 13 years ago, albeit at a much more sophisticated level.

3. Advantages of Spreadsheets in the Team X Environment

From its conception the spreadsheet was designed to solve budgeting and financial problems. So how did this tool become the primary design software of the concurrent engineering team for NASA's lead center for the robotic exploration of the solar system? Obviously the capability of the spreadsheet has come a long way since the early days of VisiCalc™, but there is more that make spreadsheets the software backbone of Team X and many other aerospace concurrent engineering teams. Concurrent engineering is about rapid transfer of ideas and information, working to the common objective of completing the design under study. To verbally communicate the team needs a common language; to electronically communicate a common language is also required – the spreadsheet. All engineers come to the team with a working knowledge of the basic operation and use of the spreadsheet and are able to support their end of the data transfer task without a spreadsheet expert's assistance.

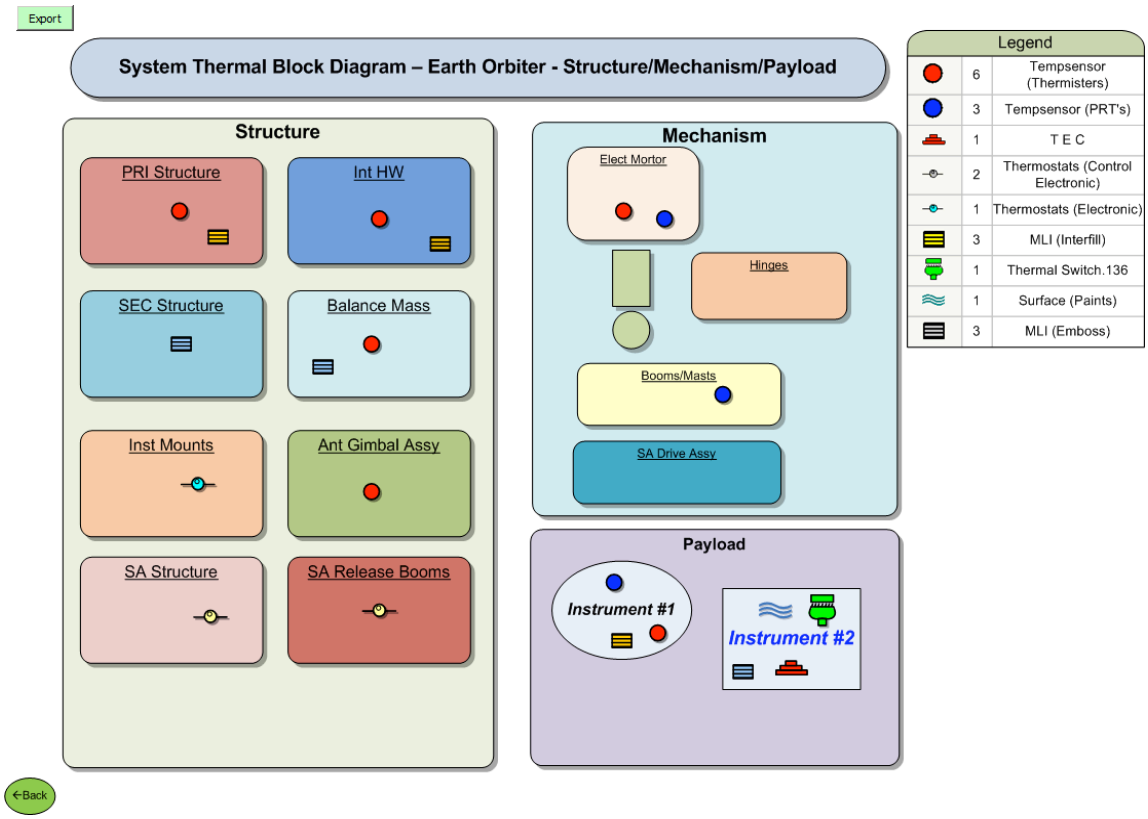
The second attribute of value is the inherent flexibility and adaptability of the spreadsheet. The software can be configured (and reconfigured) for just about any design scenario. Since studies can change in scope, objective and mission architecture on a near daily basis, the spreadsheet is invaluable in allowing the team to reconfigure its tabulations and even models. This can be accomplished often while the study is in progress. The addition of functions and internal programming languages has only increased the tools flexibility over time.

The software cost is also attractive since it is largely seen as a basic business expense (like phone services and internet access) and is procured for all employees needing a computer. Team X pays for no spreadsheet software training since all engineers are familiar from outside exposure. If during model development or update, a more advanced spreadsheet activity is required, a small number of spreadsheet experts are available and are brought in.

Many spreadsheets come as a part of a suite of tools with established interfaces and interoperability. This facilitates more elaborate tool development and template driven report generation.

4. Spreadsheet use in Team X

Beyond using the spreadsheet to move data from Chair to Chair, Team X utilizes spreadsheet capabilities in a number of imaginative ways [2]. Imbedded programming languages that now come



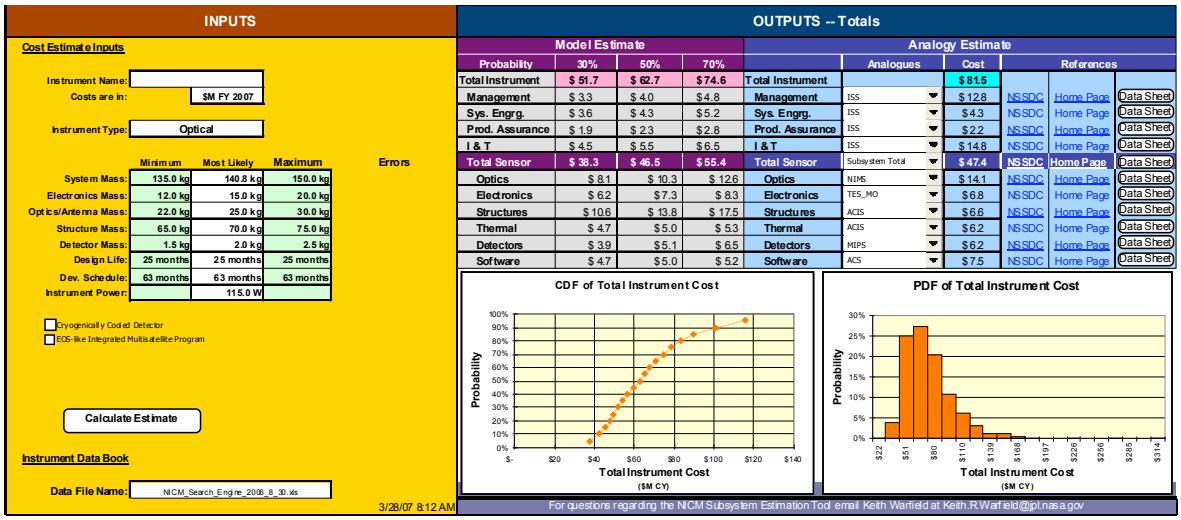
with most commercial spreadsheets have greatly expanded the capabilities of the once simple spreadsheet. Many Chairs do sizing calculations, in part or in whole, with commercial or Laboratory developed non-spreadsheet software that pass resulting data to the spreadsheet. Cost estimates can be handled as stochastic simulations using commercial or JPL developed Monte Carlo engines, which are added as new libraries of functions within the spreadsheet. This feature enables the team to calculate cost as a distribution and allows the team to perform cost-risk analysis – an emerging NASA need. One Chair has developed a drag-and-drop block diagram in a graphics program that interfaces with the spreadsheet and populates parts counts, as well as parts mass, power and cost information into that subsystem's scoping tools. Parts data are pulled from a spreadsheet-based flat file database. Team X management would like to see this type of tool expanded to all hardware Chairs since it takes over the tedious task of populating mass, power and parts lists and allows the engineer to focus on the design aspects of his job.

Many Chairs currently interface with spreadsheet stored databases or, in some cases, web-based data databases but the recent move to the NEXSYS database has put the whole team on a common, searchable relational database, and the team's management anticipates that most of these homegrown data sources will migrate to the NEXSYS system, allowing the team to standardize content, database structure and capabilities across all subsystems.

Not everything has worked out as planned. A portion of the team's work is centered on concurrent design for instruments. Different tools, people and subsystems support this work. These studies tend toward greater detail in their design work so higher-fidelity tools are more common in this environment. Few of these tools are integrated directly to the spreadsheets mostly because the instrument design work is less frequent so there is a less urgent need and fewer resources have been expended to improve the efficiency of this software environment. Data is manually transferred from many design programs to a networked spreadsheet on each chair's computer. The networked sheets are structured like the sheets used in the mission studies: all sheets interfacing to the central NEXSYS server. There are more bugs in the instrument environment, partly because it is newer than the mission environment and again, partly because the work is less frequent and is viewed as a lower priority so debugging has come as a lower priority. These issues will be resolved, most likely, before the end of the year.

5. Problems with Spreadsheets

The flexibility that makes spreadsheets valuable for the concurrent engineering arena has its dark side. Configuration management has been the team's most persistent spreadsheet problem and it manifests in two fundamental ways: tool configuration problems and study configuration problems. The former result from unrecorded modifications to the algorithms and spreadsheet interfaces by Chair members or others. Since different studies all begin from the same set of templates and different studies can be staffed by



different team members, unrecognized modifications can cause errors in the calculations since the team member operating the tool may not understand the reasoning behind the modification. Team management has attempted to deal with this by restricting the team member access to the templates, but even that has only reduced the problem; it has not eliminated it. Sometimes the few with access continue to make unrecorded changes. Additional tool configuration problems can occur after the study begins when the Chair modifies the sheets to correct an unrecorded problem or meet special needs of the study at hand. Correcting unreported problems ensures that they will go uncorrected in future versions of the templates and opens up the possibility that not all Chair members will recognize the problem and will then propagate a known error in some of the studies. Modifying the tools for the special needs of a particular study without archiving the modified tools as a separate version can result in unclear definitions of study parameters and may cause errors in interpreting the study results, especially when comparing to other unmodified studies.

The team is currently looking at some configuration management approaches used in flight software development (version control, standardized modification procedures, change control boards) to address these issues but it is not clear at present, which of these flight software practices will be agile enough to support an environment that can be running 10 or more studies in a single month [3].

The second major source of configuration errors occur when the algorithms are used correctly but the spreadsheet interaction is mishandled. This is an operational issue, not a tool issue in general; however the lack of practical access control within the spreadsheets facilitates the problem. When converging a study design point, team members will be instructed to "request and send" their sheets (i.e., update their sheets with the latest parameters available). The design is then reviewed and the option is closed. Problems occur when a member either neglects to "send" his latest data or he makes a change after the option is closed and does not inform the Study Lead or Systems Engineer. Both can invalidate the design (although this is really rare) and are not usually uncovered until after the study has ended and the customer has received preliminary cost, mass and power numbers. The SEs have instituted a policy of "resyncing the server" where by they open all the workbooks and "request and send" all the Chairs themselves after the study, but this has not always worked since some of the worst offenders have changed their inputs many days after the study has ended. Locking the sheets to changes is not practical due to the large number of sheets involved and server access restriction is equally untenable due to the large number of people and studies involved. Team X management is now using weekly product status reporting to drive behavior modification in the problem Chairs.

There are a couple of other issues with the use of spreadsheets that continue but are less severe than the configuration management. Spreadsheet documentation, spreadsheet content management, usability on non-PC computers, and consistent parameter definitions across all Chairs are a few of these other problems. Standardized documentation of the contents of the spreadsheets has been absent through most of the team's history. An understanding of the spreadsheet's architecture, its interface requirements, and its internal algorithms is required for successful configuration management. In the last year, significant progress has been made in documenting the spreadsheet architectures and interfaces, as well as the cost algorithms. Work is still remaining on the documentation of the subsystem design models. Years of no documentation and several successions of Chair leadership have left several spreadsheets loaded with obsolete material. Not knowing what is needed and what is not, new Chair Leads have simply added to the sheets without ever cleaning house making some of the sheets difficult to understand. Team X works in an all PC environment so the spreadsheets work well in the Design Center. However, many chairs do work outside of session from their offices and laptops, which many times are not PCs. Issues remain with the usability of sheets – particularly those that utilize the sheet's internal programming language – on these machines. Finally, since Chairs share spreadsheet data but control their own spreadsheet environment, a clear understanding of the meaning of that data must also be conveyed. Problems have arisen over different definitions of data being transferred between Chairs (e.g., is the pointing requirement 1 sigma or 3 sigma? Is the cost real year dollars or FY08 dollars?). A standard JPL work breakdown structure dictionary is in place to define subsystem ownership issues, but the majority of the parameters do not have documented definitions that can be uniformly enforced across the Chairs, and must be worked out orally during the sessions.

6. Verification Methods

The change in the use of the Team X products from quick early concept designs to being used to develop JPL responses to NASA's first round proposal calls

meant that all processes and procedures had to become more formal and that the accuracy of the cost models had to be certified and vetted. Previously, errors were caught in session because 20 to 30 pairs of eyes and models were interacting with the data and after the sessions the cost and systems engineers went through all the reports for consistency, requesting modifications when necessary. With the shift in criticality, formal requirements for model performance and documentation were required. Because JPL is a systems house and not an accounting house, the Laboratory's approach to improved quality is not to focus on audits [4] (although Team X rates and factors are internally audited) but on reviews and testing. The first round of implementing a formal verification process focused on the cost models has been completed [5]. A second round addressing issues with the design models is currently being planned.

A major impact of the extensive review and test process is that the elements of the organization that ultimately build each subsystem become directly involved in signing off on their respective models. This produces the organizational oversight, buy-in and ownership that are crucially needed. Spreadsheets played a major role in this process because they provide a highly assessable view of the models simplifying communication with and between experts and non-experts.

The Team X spreadsheet models were verified in three different ways: verification of organizational and legal requirements, verification of usability, verification of model accuracy.

There was independent verification that all 21 specific requirements and 8 general requirements were satisfied. These requirements consisted of items such as all cost models must use the latest approved rates and factors, estimates must have detailed documented basis of estimate, and estimates must be of sufficient level of detail that they can be compared to grass roots estimates that are generated later in the lifecycle.

Model usability and the quality of the user guide were verified by comparing estimates from two different Team X members who were not involved in model development, based on the same set of mission descriptors. These blind tests were performed simultaneously with the estimators in two different rooms so they could not communicate. The estimates had to be within 10% of each other.

Subsystem cost model accuracy was verified by comparing the model estimates to actuals from 5 to 10 historical missions. Actual mission cost data is generally scarce in NASA, and many subsystem costs were absent in specific missions where that hardware had been procured rather than built in house, so a larger sampling was not possible. Verification estimates again were done by Team X members knowledgeable in that tool in a blind test. The estimates were required to be within +/- 30% of the actuals. Since JPL is an R&D organization and there is always significant 'newness' in each mission, some normalization of actuals was permitted as long as they were well documented and provided to the review board.

In addition to the rigorous verification testing, any model update must go through an extensive integration test before they are allowed to be used during a live session.

7. Conclusions

What began as an experiment built on a mixed bag of ad-hoc sizing tools has become, in 13 years, a recognized Laboratory asset and an emulated standard within the international aerospace community for early space mission design. It is not overreaching to say that spreadsheet software was the technology that enabled this to come about.

It is no accident that every major aerospace concurrent engineering team has adopted spreadsheets as their default team design software and primary systems engineering tool. JPL's Team X, Goddard Space Flight Center's Integrated Design Center, Aerospace Corporation's Concept Design Center, and the European Space Agency's Concurrent Design Facility all use spreadsheets in a very similar manner. The combination of flexibility, built-in capability, a common understanding of the software's operation by team members, and its applicability to the team's fundamental resource estimation products make it close to ideal for the concurrent engineering environment.

There have been attempts to do concurrent design without the spreadsheet backbone. One such effort at JPL succeeded in interfacing high resolution commercial design tools to each other, but the system proved inflexible and difficult to expand and maintain. In contrast to this example, Team X has grown from simple mission studies to a wide variety of products including: instrument studies, platform studies (e.g., rovers, landers, and aerostats), complex mission architecture trades, technology trades, design reviews, direct proposal support, and cost estimation. The need to move in different tools for different studies and different customers remains to the present.

From time to time there is discussion of moving to other tools to do what is currently done with the spreadsheets (for example web based tools), but it is

hard to imagine such a change without a significant negative impact to those software characteristics that have been most valuable to the team over the years. Overall, the advantages still greatly outweigh the disadvantages so the focus remains on resolving current issues rather than moving to a new paradigm and a completely new set of issues. A current example is the recent introduction of a relational database to store all study parameters, which will improve data reuse between studies and enable cross study analysis, but works with the current spreadsheet environment rather than requiring new user interface software. In the near future, new methods to resolve the configuration management issue will be evaluated within the concurrent engineering environment, and the more successful methods will be adopted by the team.

The payoff from concurrent design has been big – possibly reducing costs by two-thirds, and a even greater return in design speed [6]. And new applications continue to emerge as the capabilities of this now not so new methodology are challenged. Rapid, parallel communication – both verbal and electronic – has been the key to its success. Without the unassuming spreadsheet, none of this would have happened. It is the catalyst that turns chaos into design.

References

[1] Thompson, E. and Davis, J. (2004), Daniel Saul Goldin, Retreived from http://history.nasa.gov/dan_goldin.html

[2] R. Wheeler, M. Adler & B. Sherwood, "The New Team X", International Astronautical Federation, 58th International Astronautical Congress, Hyderabad 24-28, September 2007

[3] Chen, F., "Advanced Projects Design Team (Team X) Configuration Management Plan", *JPL Internal Document D-45927*, 2008

[4] R.Panko. Applying code inspection to spreadsheet testing. Source. Journal of Management Information Systems, Volume 16, 1999.

[5] Anderson, T. and Rosenberg, L., "Team X/Step 1 Cost Model Development Task – FY 2007 Final Report", *JPL Internal Document*, 2007

[6] Wall, S. "Reinventing the Design Process: Teams and Models", IAF, Specialist Symposium: Novel Concepts for Smaller, Faster & Better Space Missions, Redondo Beach, California, USA, 1999