Electronic Capabilities Manager as an Engineering Process

1 Introduction

Our goal is to improve cost, schedule, and accuracy. We also expect improvements in subjective areas too, such as quality, management confidence, customer, employee satisfaction. Consider giving everyone, including customers, controlled access to all available resources, including all the resource implications.

Traditionally, we had a good understanding of our own capabilities, and more importantly understood impact of each requirement, and life cycle issues. Our representatives to the customer had every available resource and understood most implications.

Some institutions lost this basic process when project over runs became profitable. Business goals start to diverge from engineering, and responsibilities are separated from authority. Most institutions lost this basic process when the solution domain became to complex for anyone to fully grasp. Many artificial metrics have been generated and have shown great promise, but still fall short of the cost, schedule, and accuracy goals.

Herein, is a proposal to maximize your capabilities. Question to be answered: Is this a step closer to a true plug-and-play capabilities process?

Contents

1 Introduction **1**
1.1 Requirements are based on capabilities. 5
1.2 Goal 5

2 Identification **5**

3 Scope **5**
3.1 Long term goal: a complete vertical solution. 5

4 Concept: Requirements are based on capabilities. **6**
4.1 Demonstrated projects are the proof of concept. 6
4.1.1 Deliveries of the first few C130 AMP chassis. 6
4.1.2 Distributed Database Project 6

5 What could a new process achieve? **7**
5.1 Aircraft Position Example 7
5.1.1 Prototype 9
5.1.2 Level A Safety Requirements 10
5.1.3 Tight Synchronous Timing Requirements 11

5.1.4 Long Term Solution . . . 12
5.1.5 Developers Perspective . . . 14
5.1.6 Top Level Process Perspective . . . 15
5.1.7 Adding Capabilities . . . 16
5.1.8 Summary . . . 17
5.2 New Project Proposals . . . 17

6 Background **17**
6.1 History . . . 17
6.1.1 Requirements Impact . . . 17
6.1.2 In house embedded kernels . . . 17
6.1.3 Planning and Obsolescence . . . 18
6.1.4 Lost Quality . . . 18
6.1.5 Lost Support . . . 18
6.2 Significant Events . . . 18
6.2.1 Government Overruns . . . 18
6.2.2 Fall of GM . . . 19
6.2.3 Wind River sold to Intel . . . 19
6.2.4 No support for Windows 2000 . . . 19
6.2.5 Attract and Keep Gurus . . . 19
6.2.6 GPMon Loader and Test Environment for GE's PowerPC products . . . 20

7 Developing an Effective Process **20**
7.1 Process Requirements . . . 20
7.1.1 Up Front Requirement Tracking . . . 20
7.1.2 Minimize information overload and the need to know. . . . 20
7.1.3 Visibility . . . 21
7.1.3.1 Management Visibility . . . 21
7.1.3.2 Engineering Visibility . . . 21
7.1.4 Product Oriented . . . 21
7.1.5 Value . . . 21
7.1.6 Quality . . . 21
7.1.7 Paper-less . . . 22
7.1.8 Data Security . . . 22
7.1.9 Optimizations . . . 22
7.1.10 No obsolescence, dependencies, and limitation issues . . . 22
7.2 Summary . . . 23

8 Proposing using the Electronic Capabilities Manager (ECM) Process **23**
8.1 Infrastructure . . . 23
8.2 A plan for success . . . 23
8.2.1 Support . . . 24
8.2.2 Champions . . . 24
8.2.3 Plan . . . 24
8.2.4 Avionics Architect . . . 24
8.2.5 Indirect benefits of considering the ECM process . . . 25

8.3 Frequently Asked Questions (FAQ) 26
8.3.1 Will this process be Linux based? 26
8.3.2 How mature is ECM? 26
8.3.2.1 How soon can we see a demonstration? 26
8.3.2.2 How soon can we see a specific demonstration? 26
8.3.3 Is there considerable effort to setup an ECM database? 26
8.3.4 Why the strong push against third party kernels? 26
8.4 Conclusion 27

This paper re-introduces some old concepts, shows some existing examples, suggests some process requirements, then proposes a solution.

1.1 Requirements are based on capabilities.

The first step is recognizing requirements are based on capabilities. The cable technician is capable of creating cables, but requires specific connections, connectors, etc. The specifics allows the technician to report back cost and schedule.

Every resource is a capability which can be drawn upon. Each capability has implications which can be maintained electronically and linked together. Automated documentation can be used to verify every facet of the system. Existing systems lend proof to this process proposal.

Some recent significant events may help in considering the proposed process.

1.2 Goal

Our goal is to significantly improve the more complex systems such as avionics, but it requires a tightly coupled process. Improvement can be demonstrated in cost, schedule, and accuracy on a pilot product, with the necessary support and authorizations.

2 Identification

```
$Id: ecm_process.tex 28 2011-12-02 23:57:39Z ty $
$HeadURL: svn+ssh://InfiniteDelta.com/svn/proposals/ecm_process/ecm_process.tex $
```

Ty Zoerner
www.InfiniteDelta.com
Ty@InfiniteDelta.com
(616) 301-3930

3 Scope

The proposed process applies to most facets of entire and related product lines of all sizes. Marketing, systems, hardware, software, test, manufacturing, and support, can all benefit. Medium to large systems benefit the most, but it still allows small products to be integrated into complex systems.

3.1 Long term goal: a complete vertical solution.

Although this proposal for an experimental process targeting the most complex portion of an avionics system, the end goal is to support an entire avionics vertical solution. The proposed process fully

applies to multiple sites, internal and the majority of external resources.

4 Concept: Requirements are based on capabilities.

Key to this process is recognition that requirements are based on capabilities. A requirement is never accepted unless we have a capability to meet the requirement.

Specific benefits are demonstrated by actual operational projects.

4.1 Demonstrated projects are the proof of concept.

4.1.1 Deliveries of the first few C130 AMP chassis.

The C130 AMP project plan had a chassis level test plan in place, but was not available to support delivery of a few lab units. A C130 AMP confidence test would be sufficient for the first set of these deliverables. The system included 20 CPUs, 60 ARINC 429, 10 RS422, 20 Ethernet, 8 1553 buses, 20 Discretes, 10 PCI buses and one VME bus. Configuration took four days from start to verification, release, to shipping the first chassis. A final professional appearing test report was created for each unit shipped.

The quick delivery was possible by a capability oriented infrastructure. All the tests previous existed and already support four other projects. All that was needed was configuring the test to handle the specific wraps required for the chassis test harness. Most of the time was spent verifying each wrap would fail when broken.

These test capabilities where originally organized to support products. It really demonstrated the advantages of having prepared capabilities.

4.1.2 Distributed Database Project

An aircraft required an embedded distributed database which is shared among 5 systems all but one had multiple processors within it. The extra effort placed up front in infrastructure paid off considerably. A single electronically readable description file was designed which fully describes the data base, including what each system contained and the connections between the systems. A custom tool parsed and verified the file before it was allowed to be inserted into the system. Verification included band width and memory utilization. The benefits includes:

Full access to the distributed database architects capabilities without needing the architect.
Any one on the project can modify the data distribution with complete confidence.
Generated distribution source code, including the structures needed by the applications.
Accurate documentation and reference material.
Instant bus bandwidth, max response timing, and memory utilization reports.
Precise communications between each software group without many meetings.
Allowed our Systems types to support the customer with minimal support from software.
Eliminated all risk for software relating to the distribution.
Truly supported the highest operationally with multiple hardware degraded modes.
Major functionality can be added with very low cost and risk.

Note the custom utility effectively delivered the capabilities from the description file which supplied the specific requirements. The process supported an extremely high amount of changes with virtually no process overhead, which allowed a lot of changes. An average of 3 changes a day was experienced for the first year of development after this utility was created.

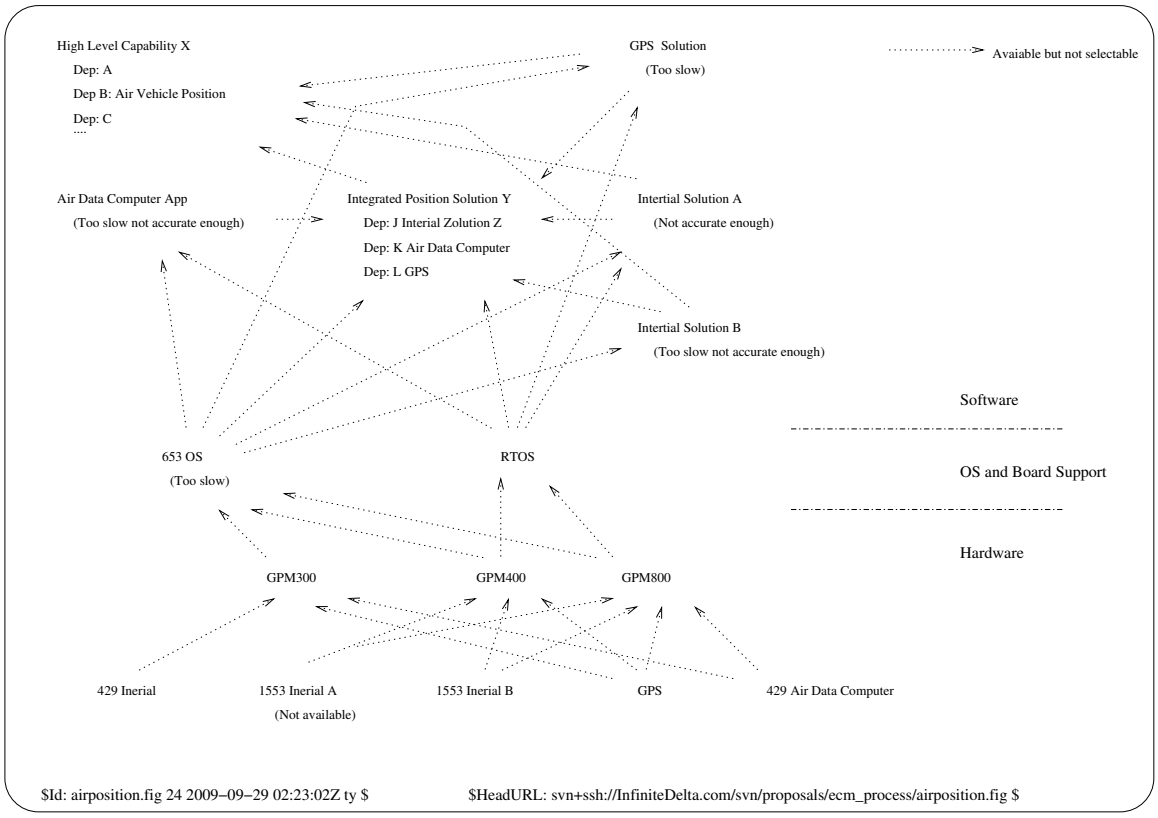
5 What could a new process achieve?

The Distributed Database Project 4.1.2 demonstrated a much higher visibility and it minimized information overload. A process encompassing a vertical oriented product line could achieve the same results on a larger scale. Consider a developer of a high level application example, which requires the vehicle's position.

5.1 Aircraft Position Example

An application needing aircraft position is an example which highlights the system interdependencies. A high level application developer can deliver some imaging capability. But the offered capability is dependent on several other capabilities including the aircraft position. This example will dive down into aircraft position dependencies. The developer does not care where the data comes from only that it meets some accuracy, format, and timing requirements.

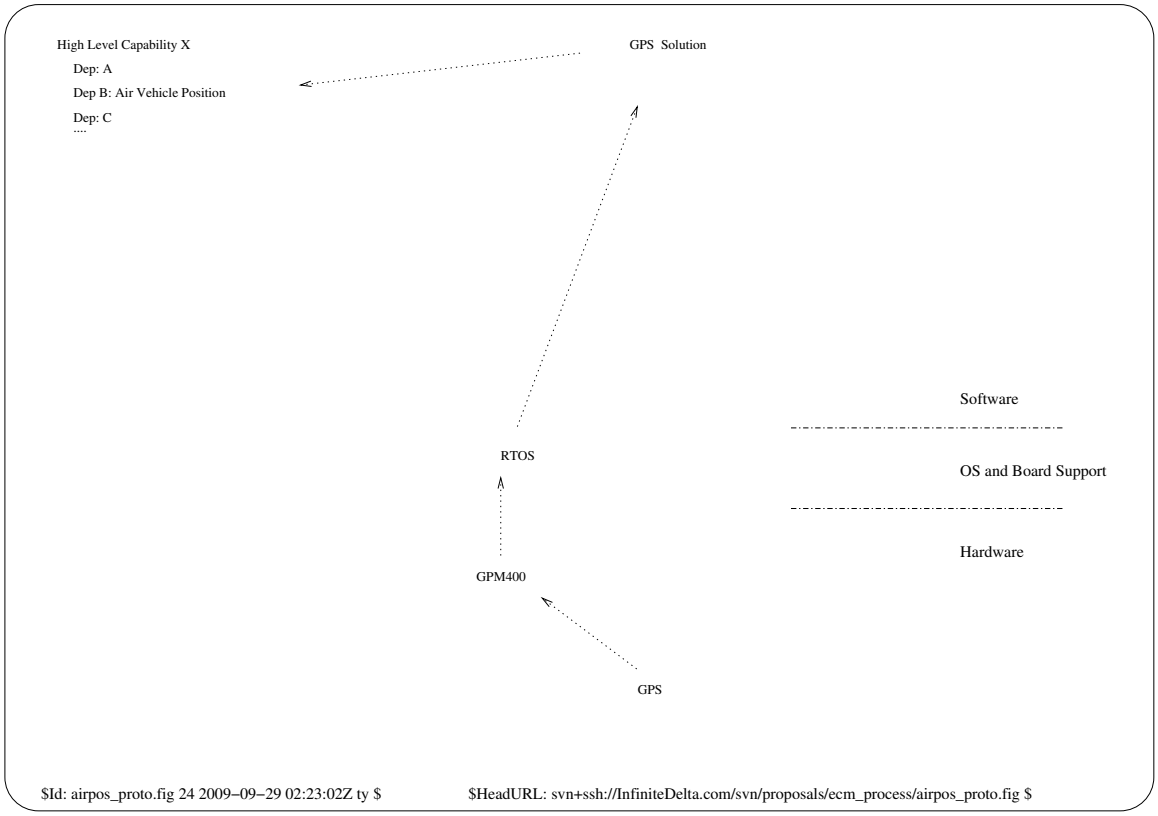
There are many capabilities in the company that can deliver vehicle position, but most are eliminated for different reasons. Those remaining have limitations of their own.



Aircraft Position dependencies.

5.1.1 Prototype

Generally a product concept is started with a prototype. So a least cost, least dependencies, and available equipment are used. The image developer chooses an existing and available piece of lab equipment to prototype their software:

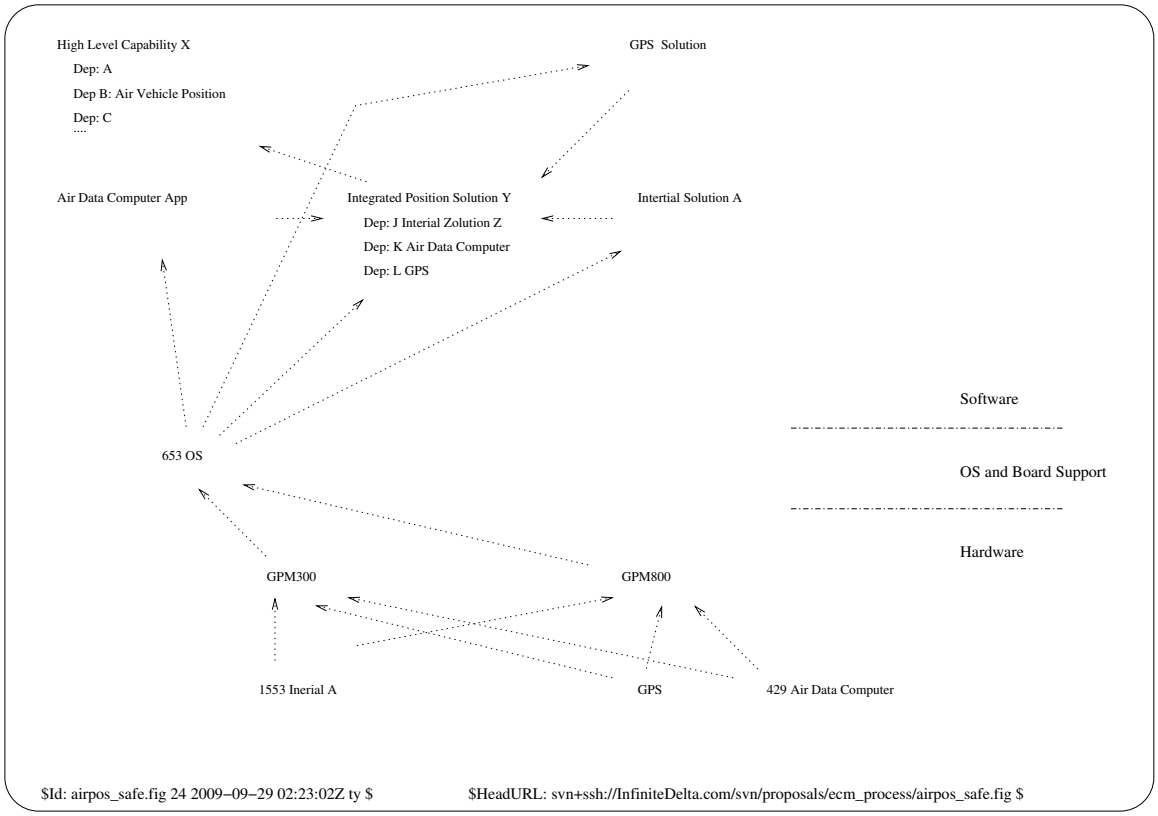


Quick Prototype.

This example shows the benefit of having lab equipment and other available resources under an ECM control. One project can rent equipment from another and automate all physical resources.

5.1.2 Level A Safety Requirements

A project may require to use the image application on a Level A certified platform:

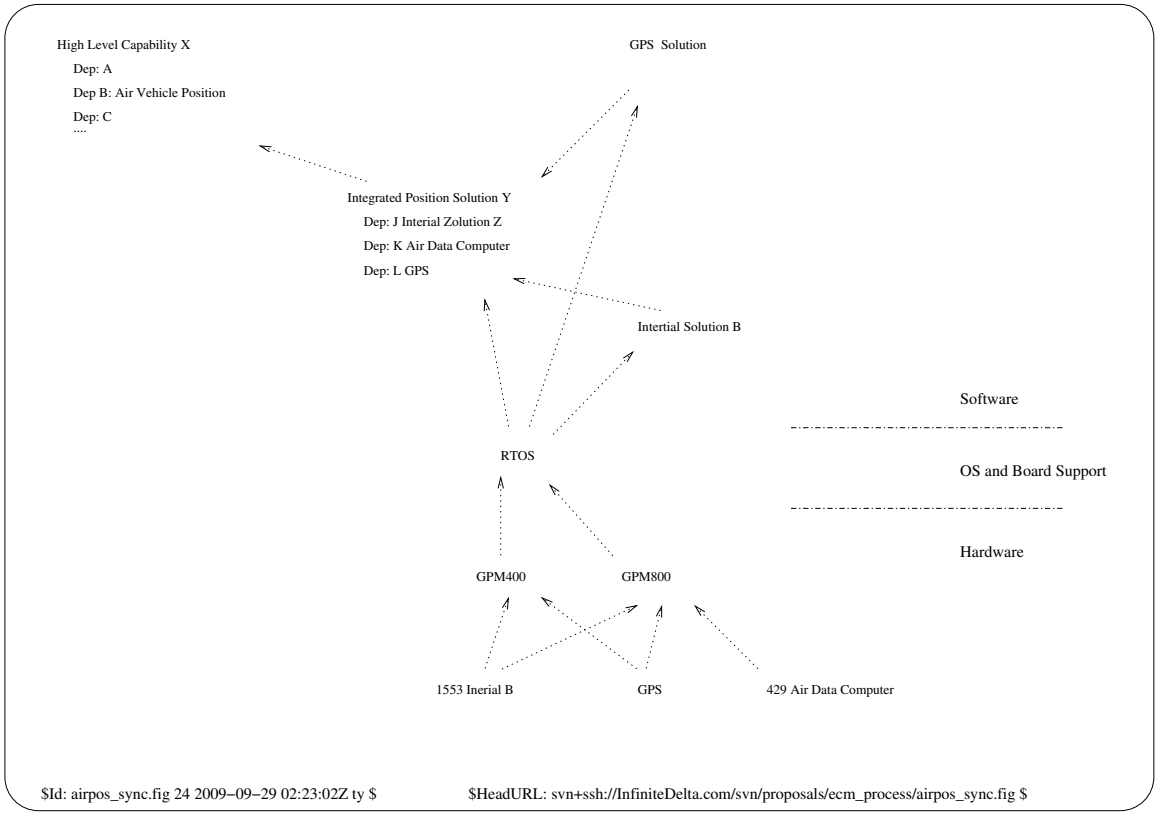


DO178B Level A Safe Solution.

This demonstrates only allowing already certified parts. A project may be halted here if there isn't already enough capabilities. Or decide to pay for the additional capabilities to be added to the system.

5.1.3 Tight Synchronous Timing Requirements

A different project may select to use the image application on a UAV for coordinated 3D image reconnaissance. This requires extremely tight synchronization requirements, eliminating the ARINC 653 OS:

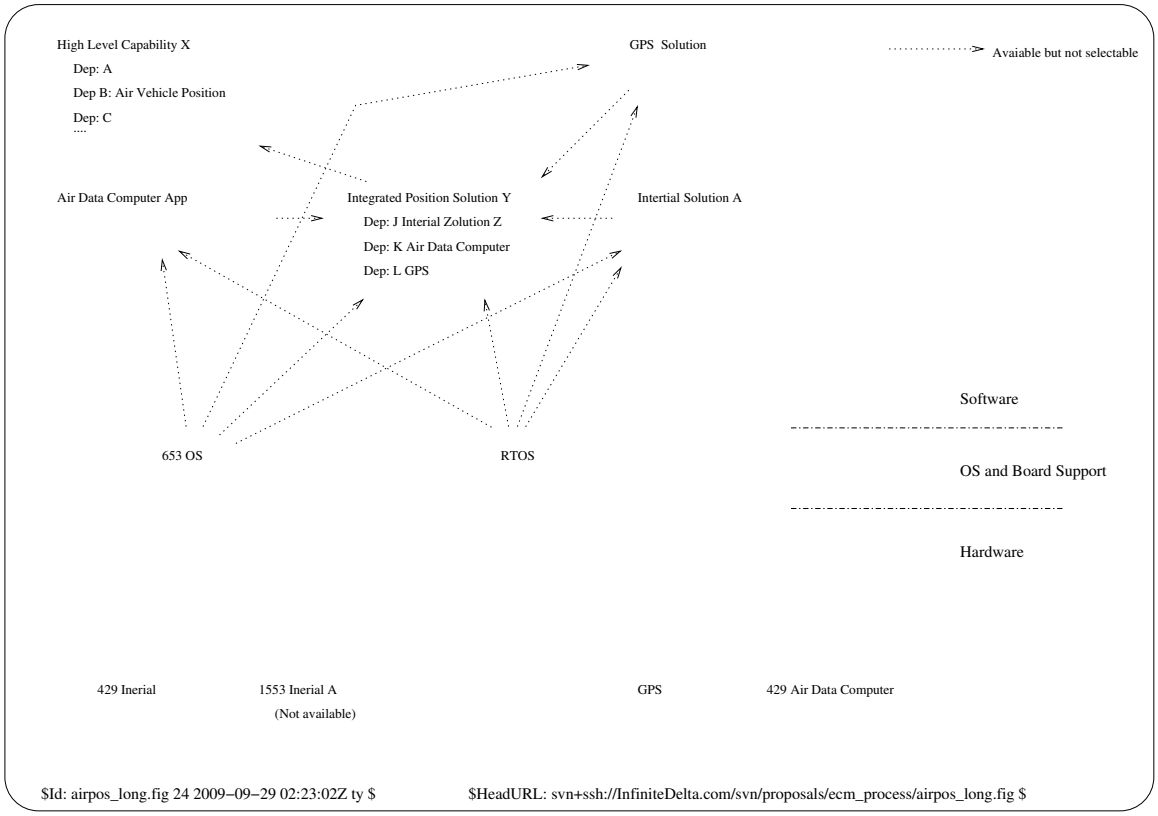


Tight Time Synchronization Requirements.

This time synchronization example demonstrates that the process can support even the toughest systems to resolve. The process takes care of issues directly and upfront.

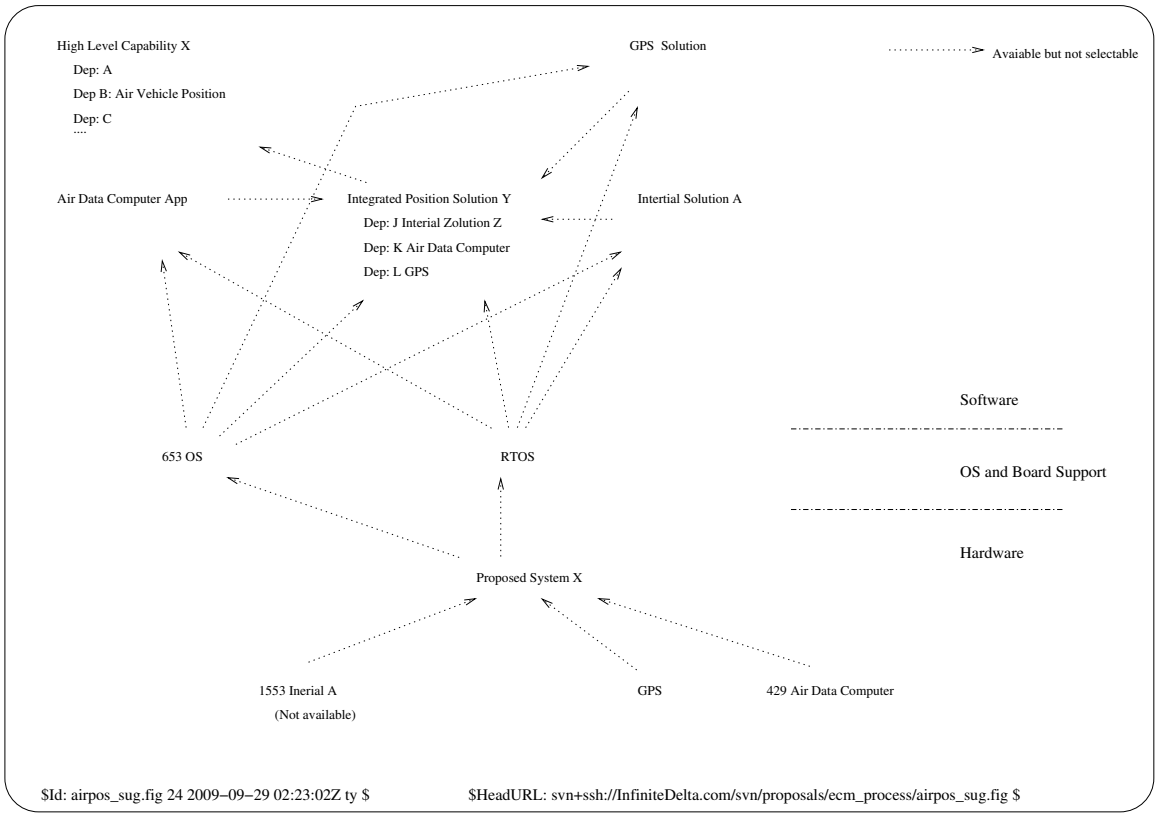
5.1.4 Long Term Solution

A different project may select to use the image application on a long life commercial product. This requires long term planning and includes parts, manufacturing, test, and long term support. A solution is not presented because there is no long term hardware:



Long Term Planning.

The missing platform needs to be created. One or more suggested capabilities are added to the system. The capabilities are added, but now includes dependencies on development time and resources.



Suggested Platform X.

Long term planning needs to be integrated into our processes. We are still supporting the Boeing 737, which first flew in 1967. Although the avionics has changed a few times, the current system started in 1993.

5.1.5 Developers Perspective

Within the current process, developers are project oriented and have access to project specific data. Using the correct position is important and leads to some project specific understanding. The ECM process reverses this concept relative to the project. The project must deliver the position to the application. A standard process would require the developer to investigate approved position capabilities. The developer could perform a search for a position capability:

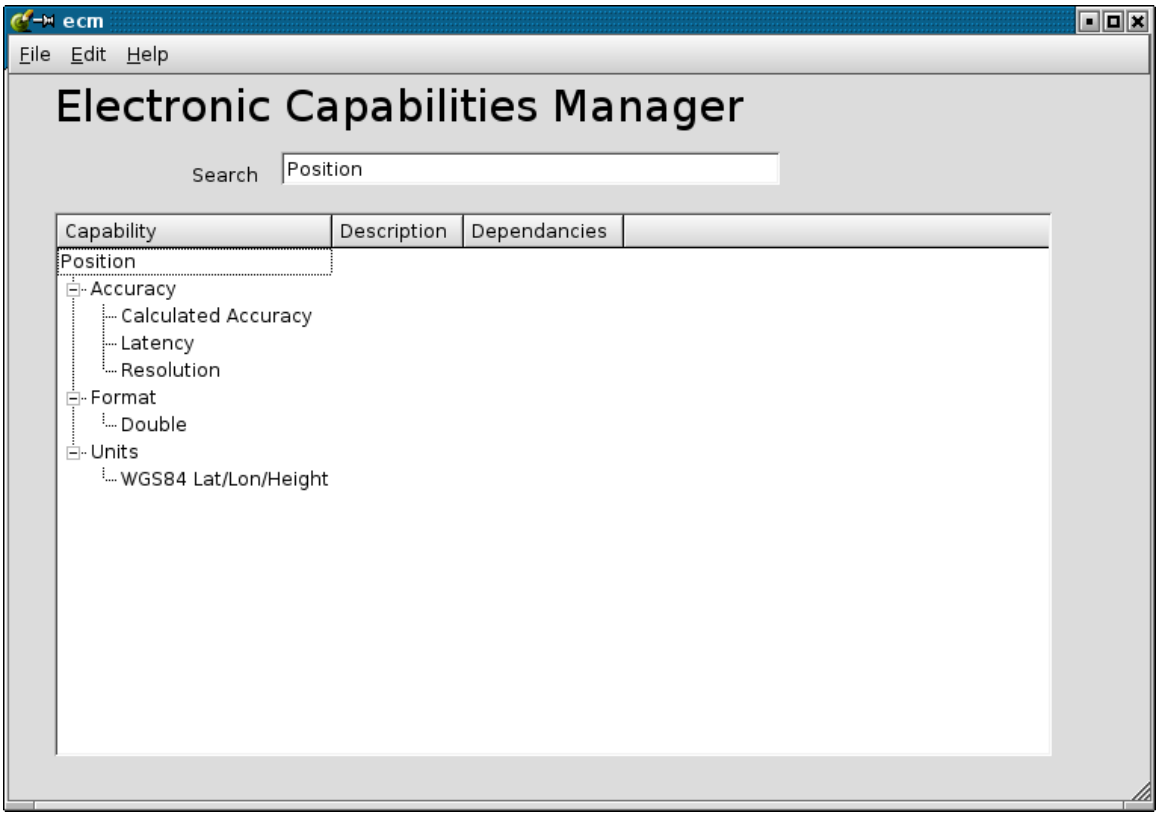


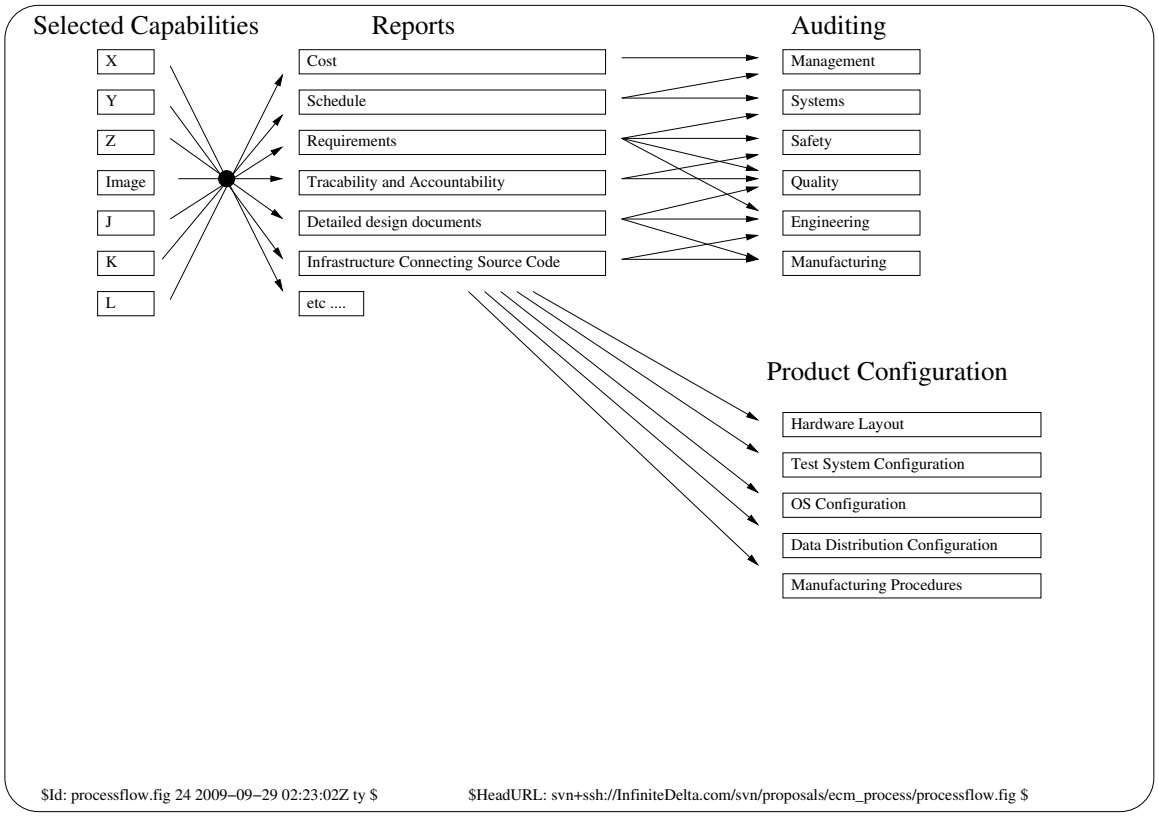
Image Developers search for Position. [1]

The quality of the application can be dependent on the accuracy of the source data. If the developer can quantify the quality based on the accuracy of the source data, then it can be passed on to those depending on this capability. It can be included everywhere in the development process. Starting from caveats in the requirements documents, all the way to the data passed between applications that may need some required performance criteria.

[1] Illustration only, not operational.

5.1.6 Top Level Process Perspective

All users and developers have the same capability search engine5.1.5. All users also can supply capabilities to the same capability search engine. However their access level and search criteria is specific to their assignment. The process flow changes dramatically from the more project oriented development. The image application is only one capability among the many selected. Each capability directly supplies data for all the reports required for the system.



Capability Process Flow.

Major change from standard processes: Requirements do not drive the system, instead they are selected and reported out of the process.

The image capability could supply a one line top level requirement of, ‘‘... shall supply an image correlation to an accuracy of ..., see requirement XXX.’’. The capability could supply a detailed requirement text, the tracability dependency data, and the required system configuration data. The image capability would also include already met certification levels.

5.1.7 Adding Capabilities

If a sufficient capability did not already exist, a request for a capability, whether it is an internal or external customer, would start the process. An easily editable and machine readable text file is created for each capability: The image application developer may create an initial capability for their application by:

```
Capability: Image X Application

  Shall support XYZ using resXY images to correlate with vehicle position.

  The accuracy shall be within accuracyX and accuracyY when condition positionX and timing Y are met.

Categories: High Level Software Application, Software Application

Requires:
  resXY: Image resolution size from resX and resY.
  resX: Image X pixel resolution, defaults to 768, but ranges from 512 to 2048.
  resY: Image Y pixel resolution, defaults to 480, but ranges from 256 to 1024.
  qualityX: Required quality, defaults to 1.0, but ranges from 0.1 to 5.0.
  embeddedEnv: embeddedEnv.ecm
  position: http://infinitedelta.com/ecm/position.ecm#WGS84
  positionAccuracy: http://infinitedelta.com/ecm/positionAccuracy.ecm
  time: http://infinitedelta.com/ecm/time.ecm
  imageData: http://infinitedelta.com/ecm/imageData.ecm
  cpuInstructions: resX*resY * 12.5 per image
  cpuMemory: resX*resY * 16 * qualityX
  imageEngineer: 600 hrs
  cableTechnician: 100 hrs

Supplies:
  imageRotation: http://infinitedelta.com/ecm/quaternion.ecm
  imageAccuracy: positionAccuracy * timingAccuracy * 3.0.
```

This example image capability is now ready for a future project. The requirements document gets its text from the Capability section. Lines containing missing references are not included into the requirements until they have been addressed.

The requires section links dependencies on data and other resources to the system. Data not explicitly supplied can default to standard values. Dependencies can include local or absolute URLs. If signed by an imageEngineer, the accountability is started for the time required for this capability.

5.1.8 Summary

Maximizing available resources to all, minimizing information overload, and the **Need to Know** are a key points to this process. The developers are product oriented. They are delivering capabilities, knowing the restrictions and dependencies will be resolved through a rigorous process.

Specific project requirements will narrow resource selections. Capability gaps become clearly apparent.

5.2 New Project Proposals

Those responding to RFPs would now have full access to company resources contained in the new process. More effective bids can be delivered containing much more accuracy, better tracking, and less cost. Even potential customers could combine capabilities for purchase. Far more bids can be made and supported.

6 Background

Process is the most critical issue to most companies containing complex systems. The background behind the processes lead to successful processes.

6.1 History

Lessons not learned, are really at issue, and why history is so important.

6.1.1 Requirements Impact

I started LSI in 1985 under Dick Corey who was our software lead. He had rejected an idea I had because its impact upon manufacturing. When asked, ‘‘Does all leads require a full understanding of the product life?’’ He was unequivocal about the importance of tracking impact from project, back to product, then back again to associated projects. It was the LSI corporate policy practiced by all, and worked well in its proactive environment.

6.1.2 In house embedded kernels

Miss information has been spread about the costs and schedule of in house embedded kernels relative to some third party kernels. Confusion was created because of the Board Support Package (BSP) was included into early cost center called ‘‘Kernel’’. A code analysis shows rare code changes to the kernel, events, context, and internal communications. Where as what we now refer to as BSP does require considerable work even on those third party systems. See the F4, P3, and FSAS source code library for details.

How many changes where made in the 737 kernel through its long life? Can it still be supported?

6.1.3 Planning and Obsolescence

The LSI product oriented process placed considerable importance on obsolescence. A simple concept that allowed us to supported a 1970's ARN101 system in 2006, but had to drop a 2002 Smiths PowerPC product because of pSOS obsolescence, and force to start a VxWorks (Wind River) PowerPC BSP support over for the 767 tanker, which is now owned by Intel.

All programs using the third party pSOS RTOS must be abandoned or rehosted to a new RTOS, such as NIMROD.

6.1.4 Lost Quality

Engineers distance themselves from forced insufficient third party solutions. Even the most proactive not directly impacted become defensive. By definition ownership is lost, and the work arounds begin. Ignored predictions of failures on portions of a project, causes developers to swing defensive. Actively ignoring these predictions has even led to project abandonment by individuals. Even the lawyers, avoiding liable issues, are asking for no content e-mails.

LSI/Smiths had many groups and they where organized into empires. As annoying as the empires where, removing a sense of ownership is far worse. Genuine support comes from those people who developed a resource and wants to see it used properly.

Even the test and diagnostics support had been designed into products, because the BSP group where responsible for supporting software and all test phases.

6.1.5 Lost Support

Although some sections of this paper seems direct and almost antagonistic towards some processes, the end goal is to support all aspects of product development, including management, financial, developers, and manufacturing, with highest level of visibilities. The first requirement for all, even the processes them selves, is to support their customers.

6.2 Significant Events

Some recent events may help align corporate and engineering motives into a unified process.

6.2.1 Government Overruns

The first major event is a government's interest in changing the core policy regarding cost plus programs and paying for project over-runs. While the government supported over-runs, businesses only needed to throw more people at under bid projects. Many projects duplicated each other efforts since there is genuine overall cost saving motivation.

Projects are lost to companies who underbid. Initial WAGs (Wild Guesses) are preferred to performing accurate planning and analysis to prevent project loss to competitors. Hence, engineering

lost corporate support, solely on its relevants to the underbid business model. Todays individual management status is now more based on the number of people instead of the criticality of function. LSI was the slowest growing military defense company during the Ronald Reagan era, because it was slow to adapt to the under bid WAG policies. [2] The current number and severity of project over-runs, requirement creep, schedule delays, demonstrate our adapting to the government over-run policy.

Before the cost-plus policies and to be anticipated when over-runs are no longer supported, it is critical to get the right people in the right place, and listening to them. The goal is to support these critical people on the their issues, and to minimize the to burden them. Expect this new government policy to help align the goals of corporate and these critical people. Only engineering can truly guide and control engineering for the best cost, schedule, and quality for a product life cycle.

6.2.2 Fall of GM

The fall of GM and Chrysler has shaken financial people everywhere. Many loyal GM followers predicted major problems when Roger Smith, a financial type and GM's CEO 1981-1990, severely cut core engineering. An important side note: The 1990's, GM placed an MTU engineer as head of the Corvette line. The Corvette product has since made a strong recovery.

6.2.3 Wind River sold to Intel

The purchase of Wind River Systems Inc.[3] by Intel, which is a competitor of the PowerPC. Does any PowerPC based company want to prepare for potential obstacles, costs, or complete lack of PowerPC support from the Intel owned company. Typically engineering, already has the majority of pieces to eliminate this risk, but only if we seriously address some misconceptions.

6.2.4 No support for Windows 2000

The Windows 2000 support including new security updates and security-related hotfixes terminated on July 13, 2010. [4] How much of your infrastructure is based on non-supported software or equipment.

6.2.5 Attract and Keep Gurus

Are you able to attract and keep gurus? Do you have a recent exiting of several guru level developers? Loss of gurus can be very demoralizing for many areas of engineering. The lost value is agreeably not easily calculable, but it can really resonate through facilities. It exacerbates poor moral, when the majority of the exiting gurus can be easily preventable.

[2] 50 Times Around the Sun, F15 Power Distribution.
[3] http://en.wikipedia.org/wiki/Wind_River_Systems
[4] http://en.wikipedia.org/wiki/Windows_2000

6.2.6 GPMon Loader and Test Environment for GE's PowerPC products

A General Purpose Monitor (GPMon) is on several GE product lines (C-130, Tanker, E2D, Toshiba, and 787), and is embedded within the lab stations and products. GPMon was not planned, formal requirements where not written until after the product was operational. Many consider GPMon as an after thought, or just an optional product capability. Its initial 120 hour protocol development for host and target, lead to its use on many major programs. GPMon alone demonstrates the need to move to good product planning including addressing all the requirements and dependencies. GPMon's Micro Manager was selected for 787 robustness test because it supported the teams needs.

7 Developing an Effective Process

The proposed process is re-incorporating lost, pre cost-plus, process requirements. It proposed process uses recent methods for specific requirement and capability tracking. It must be considered experimental since it is neither currently established or coincides with current processes. Like any product, the process must be prototyped, demonstrated, reviewed, robustly tested, audited, before it can made available for most products. The key lost process items include visibility, product, value, support, and quality.

As with any product, the process has requirements which must be accepted. The second step is to review some proposed designs and their concepts against the approved requirements.

7.1 Process Requirements

This section covers the requirements for the proposed process. Since the process itself deals with project requirements which is confusing. For the rest of this document, the term ''requirement'' shall only refer to the process management of requirements. The term, ''shall'', refers to the process itself.

7.1.1 Up Front Requirement Tracking

The process shall support up front requirement tracking to capabilities and resources, including their dependencies. An example: A developer creates a capability that can be used to meet a specific set of requirements. The developer also supplies all the dependencies such as caveats, limitations, risks, and costs. Should a project select that capability, all the dependencies are included. Meetings can be minimized since the requirement description, schedule, costs, and risk are already included. Full accountability has also been committed right to the developer and the dependent resources. Should resources become delayed or unavailable, immediate feedback is possible, including its impact scope.

7.1.2 Minimize information overload and the need to know.

This process shall minimize information overload. Minimize data mining can resulting in large savings in effort and far better data security protection.

7.1.3 Visibility

The process shall give full visibility into requirements, designs, changes, costs and schedule are critical to all systems. Medium to large systems shall have an effective process tools to relay these aspects. Selective information targeted for the developer is all that is needed. Information overload shall be minimized, without compromising management and engineers visibility into the system state.

7.1.3.1 Management Visibility The process shall support the highest management concerns of schedule, costs, and risks. These concerns apply for all phases of a project including initial concepting and bidding. The anticipated anti-cost-plus policy is expected the change management motivation to an accurate process make proper decisions. The process shall report schedule, costs, and risks on a continuous dynamic output for all to monitor.

7.1.3.2 Engineering Visibility The process shall support engineering's content need. Specific requirement content shall be delivered in a human and machine readable context. The process shall support initiating locally developed tools to process these requirements, and report back detailed status information. Detailed information and specific dependencies are critical to an automated process reporting.

7.1.4 Product Oriented

The process shall serve projects by being product oriented. Although a little more difficult to cost out, but a far greater project support can be achieved. See the first C-130 AMP chassis confidence test example4.1.1. Many projects had to adapt other project solutions for the same product line. A product oriented process could support short schedules without the loss of any quality, which is typical in plagiarized projects.

7.1.5 Value

The process shall support dynamic value assignments on a variety of levels. Value to the company varies from project to project, and often changes. Value within projects different aspects value varies. Value can be added from specific managers, contracts, sales and support individuals can be added. Value varies based on profit, costs, bonuses, penalties, liabilities, schedule, safety, recurring versus non-recurring, and project life. The process shall reflect this on the capability selection and through schedule prioritization.

7.1.6 Quality

The process shall support a quality sign-off system beyond the required signature, but every aspect. Everyones reputation of involvement and where involved shall be tracked and used for performance evaluations. The process shall also support review issue tracking to encourage proactive issue resolutions.

Quality suffers when process prevents proper solutions from being implemented. Further quality is lost when process drains gumption by slowing or preventing proactive actions. Improper attitude, tools and environment is the leading cause of errors[5]. Forcing responsibility of a third party systems when a clearly better in house systems and solutions exists is an especially deadly attitude killer.

7.1.7 Paper-less

The process shall support a full paper-less system, complete with electronic digital signatures and tightly controlled automated sign-offs where possible. Just repeating a papered process does not lead to the savings a paper-less system can give. A re-evaluation of process steps is typically required. An asynchronous (non ordered) solution where possible can often lead to large cost and schedule savings.

7.1.8 Data Security

The process shall support many levels of data security, including automated security auditing.

7.1.9 Optimizations

The process shall support a ''what if'' investigations. Example, someone taking a day or two off, could see the effects seen by management. A manager re-prioritizing a resource, could see the effects. The process shall support automated contingencies, and searches. Anticipated risks could be prepared for and enabled when conditions are met. The process shall support quick analysis of impact upon certain scenarios, which effecting the project and those monitoring the system.

A product engineer can prepare for a contingency should part of a schedule slip.

7.1.10 No obsolescence, dependencies, and limitation issues

The process, itself, shall only use an open source system `http://www.opensource.org/docs/definition.php` available to all potential vendors and customers. The process content shall use a transparent [6] text to support core input contents and relations. The communication between process systems shall use an openly defined protocol, even when the data itself must be encrypted to protect data security.

Processes have extremely long life spans and can not be limited to planned obsolescence practices. To many projects could not be supported because of dependencies that no longer exist.

Ideally only a process developed under The Debian Free Software Guidelines (DFSG) [7] should be considered to allow a free and automated distribution of the process to all vendors and customers.

[5] Zen and the Art of Motorcycle Maintenance, by Robert Pirsig. Can you define quality?

[6] Transparent: machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors

[7] http://www.debian.org/social_contract#guidelines

7.2 Summary

Defining, reviewing, and accepting requirements for a process is the critical first step. Only then can we consider a rebuilding an effective process.

8 Proposing using the Electronic Capabilities Manager (ECM) Process

Moving entirely to an electronic system for a product for its entire life-cycle including the engineering and manufacturing process (paper less). A paradigm shift to a **Requirements are in terms of capabilities.** model. Consider capabilities as the core of a product process, whereas requirements are core to specific projects. An Electronic Capabilities Management (ECM) tool maybe considered. The ECM process is capability centric with tight coupling to requirements. All actions revolve around the capability. Anyone needing a capability would perform a search for a specific capability or a class of capabilities. Capabilities have dependencies on other capabilities which are locally owned and controlled. Benefits of Electronically Maintained Capabilities (ECM) would include:

Requirement options can be presented and verified up front.
Simpler association of capabilities to customer written requirements.
Requirements documents could be automated and accurate.
Electronic dissemination in the form required for the capability.
Many critical aspects of project tracking can be automated.
Instant reporting of issues.
Project issues are easily prioritized and visible based on impact.
Minimizes risk and need to know for each aspect.
Gives systems and management higher confidence in projects and products.
Removes the human from several key points prone to error.

8.1 Infrastructure

Infrastructure is key to any tool of this nature. The above distributed data tool directly created the tables located in each computer which drove the entire system. The embedded design is based on a system architecture capable of supporting this custom tool. A product oriented architecture including the embedded design is required. GPMon is an ad-hoc test system and an example of this infrastructure which supports five projects, because it use a capability oriented infrastructure.

8.2 A plan for success

I truly believe this process is as effective as claimed in this paper. I am open to any mechanism that allows it to succeed, anywhere from a simple support contract, to being fully hired in as an Avionics Architect. The key to success is getting the required support, champions, and plan. Please consider my reputation includes low cost, but very effective solutions.

8.2.1 Support

Many new policies and IPs shall be used, but will reflect the intent and goals of the current policies. Major process flow changes are expected. These will be developed closely with those directly effected.

8.2.2 Champions

The development of specific processes requires champions in each area. Champions are experts who are also willing to attempt a significant process change. No one, except me, needs to jeopardize their career. Everyone needs to be supported, including being able to select their involvement commitments.

8.2.3 Plan

This process must start as an experiment, gathering maturity. More than any other product, this process needs to be robust tested, verified and audited. Premature release into mainstream must be guarded against. The power and flexibility of this process also can lead to abuses, which must be monitored.

8.2.4 Avionics Architect

If an Avionics Architect is desired, regardless who is chosen, some considerations to include are:

Architect Objectives
Create and support a production oriented infrastructure.
Demonstrate clear and up front requirement tracking and the associated dependencies.
Show cost, schedule and quality benefits over existing processes.
Work with all organizations to automate the engineering and manufacturing processes.
Able to support a digital signature and encryption schemes into the automation.
Include manufacturing within the product and project life-cycle objectives.
Integrate vendor capabilities into the available resources.
Eliminate, minimize, and track dependencies on obsolescence.
Reintroduce objective accountability with specific incentives.

Architect Position Requirements
Has clearly demonstrated product infrastructure design skills.
Understands the relation between products and associated projects.
Experience with moving from paper process to a paper-less process.
Worked in all major areas including Hardware, Test, BSP, Manufacturing, Software, IT, and FMS.
Has demonstrated ability to expose core issues and common misconceptions.
Has clearly demonstrated tool generation, support, and requirements checking methods.
Has clearly demonstrated developing automated test systems,
with generated artifacts.
Understands core intent for required key organizations, including Marketing, CMO,
Quality, SQA, Security, IT, and Logistics.
Has clearly demonstrated a proactive, pragmatic (low cost), and supportive attitudes.
Has clearly demonstrated optimization and effective resource usage.
Has authored multiple custom embedded RTOS kernels including context switches, events,
interrupts, communications, and driver support.
Has clearly demonstrated work around solutions when support was not available.
Understands PowerPC Machine Check/Interrupt issues.
Has clearly demonstrated diagnostic requirements for product life cycle
phases, especially environmental.
Has experience with many different engineering processes.
Has support from key areas required for a product oriented process.

The key here is to get someone who really cares and willing to accept performance related evaluations.

8.2.5 Indirect benefits of considering the ECM process

There are many benefits to considering the ECM process. Specific lessons learned can be a value into the current processes. However, the best reason to even consider a new process is to motivate everyone. Those opposing the ECM process will be motivated to optimize the current system and find issues with ECM. Both ECM and the current process benefit. Those supporting ECM have an opportunity to optimize the process specific for their needs.

Many times managers at different companies have taken advantage of my reputation of getting jobs done, by suggesting using my services to backup current systems being developed. The implied

completion motivated developers for a more satisfactory solution.

8.3 Frequently Asked Questions (FAQ)

8.3.1 Will this process be Linux based?

Net necessarily, unless I'm the architect, then much of the system will be Linux based. Obsolescence can not be avoided, but its effects can be minimized. Also getting a coordinated customers and vendors solution requires a public process, compatible with proprietary systems, and with a solid distribution scheme. See:

`http://www.infinitedelta.com/going_linux.pdf`

`http://www.debian.org/social_contract`

Note, a web front-end can support much of managements needs. A Window's hosted X-Server allows full access to the ECM tools.

8.3.2 How mature is ECM?

Concepts only with several prototype starts.

8.3.2.1 How soon can we see a demonstration? There are several generated renewable reports available now, which include residential geothermal heating/cooling, wind, and solar data. They show specific data for the area and how specific equipment would perform on a specific home.

8.3.2.2 How soon can we see a specific demonstration? A series of demonstrations is recommended, each intended to gather more support, direction, and budget. Start with a trivial demonstration can be performed within six months, with little support, a small budget. A simple demonstration can be performing within twelve months, but requires much more support and an actual budget. An effective demonstration is expected within 18 months.

8.3.3 Is there considerable effort to setup an ECM database?

Yes, but it is directly proportional to the number and complexity of the capabilities being represented.

8.3.4 Why the strong push against third party kernels?

I've worked on over half dozen RTOS', four of which where custom, two of those I created myself. On all the projects the kernel and its core components is tiny and remains nearly unchanged throughout its life. There is no expense there. However, the Board Support Package (BSP) on the other hand has been expensive. Early estimates of third party kernels did not include the custom BSP cost that is normally required. These analysis where miss represented so badly that third party kernels

actually end up costing much more than the custom ones. Specially when you add that our early BSPs included full test support. The pSOS RTOS is now obsolete because of Wind River,,,, and Wind River has now been bought by Intel....

Frustration on this point comes from several issues:

Continuing battling this miss information.
LSI/Smiths/GE has almost a dozen custom kernels.
The PowerPC architecture being so clean and easy to work with.
much easier then most of processors used on those other custom kernels.
Full up process protections are trivial to setup.
PowerPC has a built in tickless real time support.
PowerPC also has a built in trace exception support, ideal for level A certification.
The entire GE Micro Manger kernel is only a few K of memory and far superior to VxWorks 5.4 and 6.0.
The GE Micro Manager scheduler super trivial, only 40 lines of Ada code.
There is a stigma or some kind of black-magic phobia around RTOS kernels.

I admit there are a lot of registers to save and restore for a context switch, but there is no complexity. I also do not know about all the DO178B Level A certification requirements, but this is the best processor I've worked on yet.

I've created a public domain paper targeting certifying an prioritized preemptive avionics RTOS: `http://infinitedelta.com/wp/avionics_rtos.pdf`

8.4 Conclusion

GPMon and its derivatives have been an important part of every major PowerPC project. It demonstrates what an product oriented infrastructure can do with little cost, very little planning, and only a capabilities orientation. Imaging what could be done with some planning, full up front requirements checking, dependencies enforced, full company resources available, and information overload minimized. The best proof of concept of this process is a Micro Manager based PowerPC product including the product test and life-cycle artifacts.

This paper assumes corporate is genuinely motivated to get more competitive in the complex avionics system market. This paper attempts to target process problems by addressing some key causes of those issues. It suggests some specific requirements to be placed upon the these processes. This paper introduces the ECM process, concepts, shows some examples, and proposes a few ways which to start. A full detailed description of the process concept is on the web at:

`http://www.infinitedelta.com/engineering/intro.html`