

• Rapid Application

Behavior Driven Development

Structured Programming  
Model Driven Development

Java  
Process-Oriented Programming

Waterfall Development

Requirements Driven  
Development

Test Driven .Net

SPIRAL DEVELOPMENT

Development

OBJECT ORIENTED DEVELOPMENT

# Report Documentation Page

Form Approved  
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE <b>MAY 2011</b>		2. REPORT TYPE		3. DATES COVERED <b>00-00-2011 to 00-00-2011</b>	
4. TITLE AND SUBTITLE <b>The Problem of Politically Driven System Development</b>				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) <b>PajamasandSoftware LLC, New Haven, CT, 06515</b>				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT <b>Approved for public release; distribution unlimited</b>					
13. SUPPLEMENTARY NOTES <b>Presented at the 23rd Systems and Software Technology Conference (SSTC), 16-19 May 2011, Salt Lake City, UT. Sponsored in part by the USAF. U.S. Government or Federal Rights License</b>					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT <b>Same as Report (SAR)</b>	18. NUMBER OF PAGES <b>21</b>	19a. NAME OF RESPONSIBLE PERSON
a. REPORT <b>unclassified</b>	b. ABSTRACT <b>unclassified</b>	c. THIS PAGE <b>unclassified</b>			



# The Problem of Politically Driven System Development

---

Donald L Johnson  
New Haven CT  
Software Developer (Ret)  
[PajamasAndSoftware@live.com](mailto:PajamasAndSoftware@live.com)

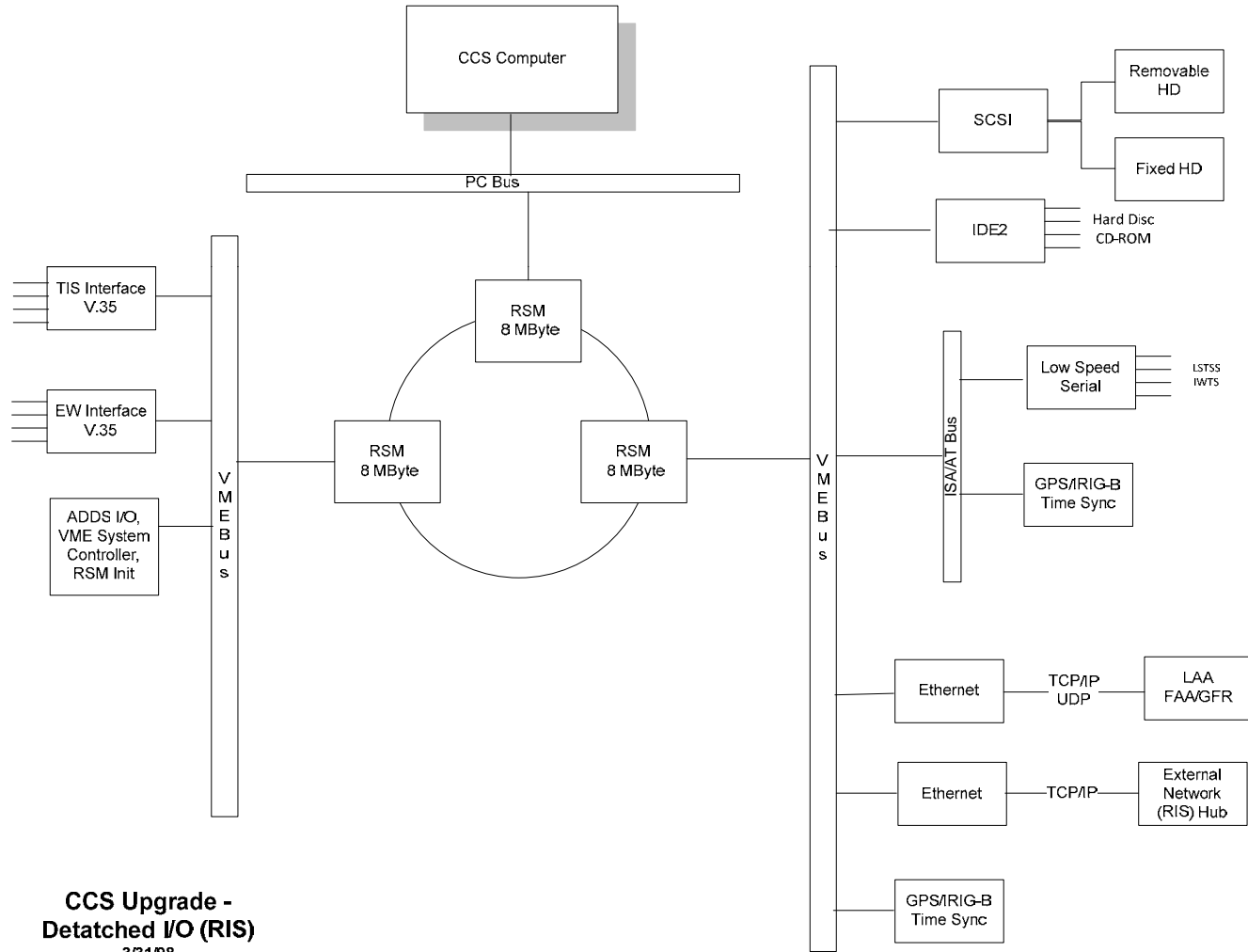
# Politically Driven System Development (PDD)

- *A method of system development which ignores proven principles of good engineering and business practices, and comes close to, if not crossing, ethical and/or legal boundaries.*
- *The objectives of PDD are often aimed at preserving existing products or organizations, but will often fail because they run counter to good engineering and business practices.*

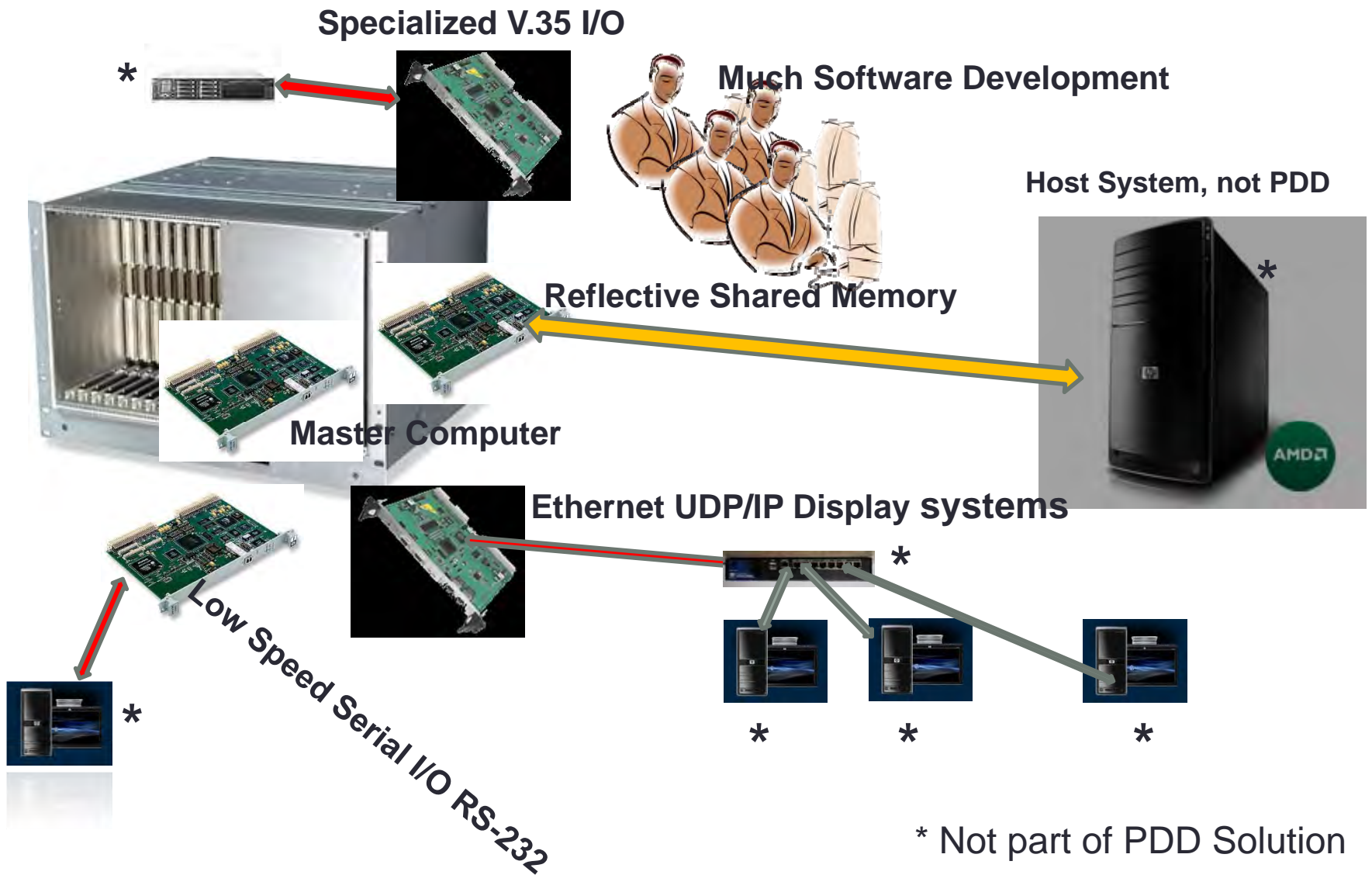
# A Case Study - Background

- A large, complex and highly successful legacy system was being replaced by an entirely new system.
  - With a totally new contractor
  - With a totally new software design and languages
  - With a totally new hardware design
- Therefore the existing Software Support Activity (SSA) was threatened with obsolescence
- The new system was still required to interface with legacy subsystems, some of which the SSA maintained, and others not
- A conscious and overt decision was made by Government SSA management to carve out the I/O interfaces as the exclusive domain of the SSA
- At the same time, an ongoing effort was underway to port a subsystem of the legacy system from mini-computers to a PC

# The PDD Solution



# The PDD Solution



# Problems with the PDD Solution

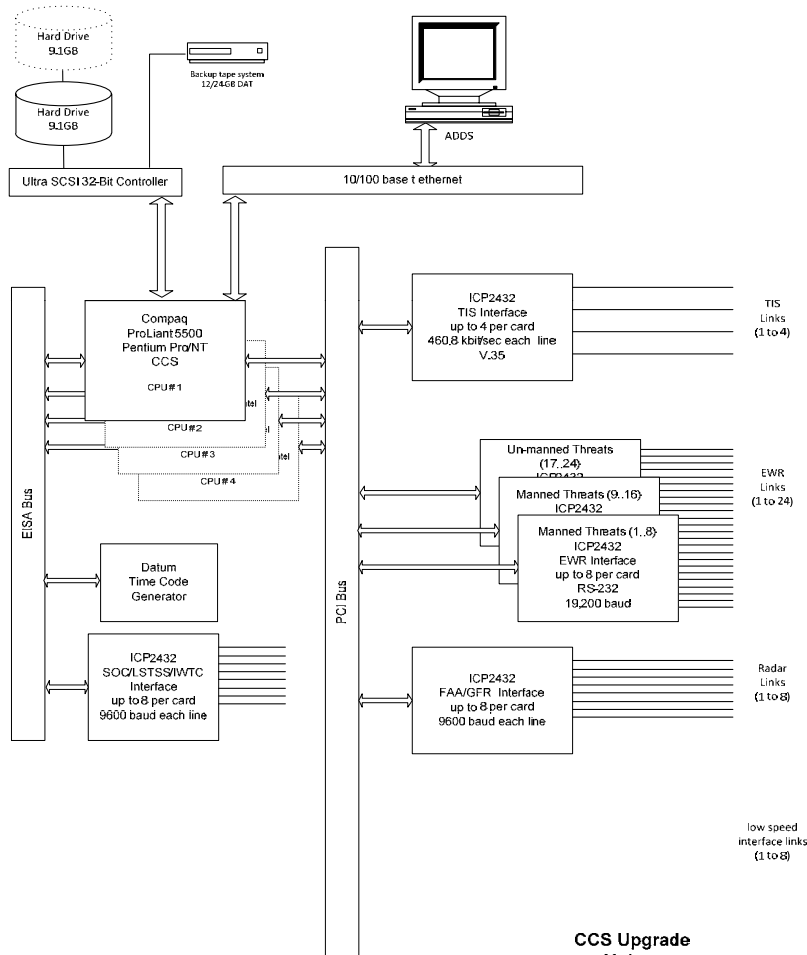
The problems I saw included:

- Orders of Magnitude More Expensive
- Required much unnecessary and complex hardware
- Required much unnecessary and complex software development
- Complicates integration & test with the new replacement system
- Future software maintenance costs increase because of unnecessary system and vendor fragmentation
- End user operations complicated by additional user interfaces
- After three years of development, not one single byte of data was passed to the target system

# Attempts to Change Course

- From the outset, I attempted to change the direction of the proposed PDD product
  - Verbally at first, to no avail
  - A number of memos were written to management
  - Studies were done showing relative cost and schedule impacts
  - Finally, after several years, a meeting with local SSA management and our Navy Sponsor (NAVAIR) provided me the opportunity to present my case to the Navy half of the joint Navy/Air Force management team
    - The NAVAIR sponsor asked for follow up, which was provided, but no action or feedback of any kind was ever received from NAVAIR
  - On one occasion I was told to “... ***sit down and shut up, this is a political decision ...***”

# Alternative Solution



**CCS Upgrade  
Yukon  
Using SimpacT Tightly  
Coupled I/O**  
D.L. Johnson 3/20/98

This solution was based on a proven architecture that had been in place in the legacy system for close to 30 years.

This solution is entirely contained within a single PC chassis, and is co-resident with one of the subsystems.

# A Simplistic Solution Rejected by Management

A simple off the shelf PC

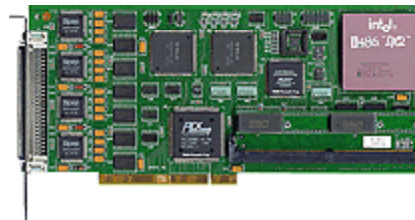
**Low Speed Serial I/O RS-232**



**Ethernet UDP Display systems**



**Specialized V.35 I/O**



# Advantages of the Simplistic Solution

- Orders of magnitude cheaper than the PDD system
- Hardware is mostly inexpensive cables (RS-232, Ethernet)
- One specialized V.35 card with firmware required
- Most I/O software required only a port from pre-existing Unix solutions

# Consequences of the PDD Solution

- After three years of non-performance, The Air Force CoSponsor demanded changes, and threatened to pull out of the SSA. This directly threatened the very existence of the SSA
- The PDD product was canceled
- Fortunately the technical staff had a backup design in place along the lines described above, and was able to produce a working product within months of PDD cancellation

## Consequences (Cont.)

- Unnecessary cost and schedule risks
- Staff moral is compromised
- Potential and real loss of staff (yours truly included)
- Scapegoating occurs after the failure of PDD
- Organizational integrity and reputation is jeopardized
- Potential for loss of new business
- Potential for entire organizational shutdown
- Potential for legal action and criminal convictions
- There is little or no accountability in the case of PDD failures
- ***Wasted funds are not available to the war fighters***

## PDD is Professionally and Emotionally Difficult for the Entire Staff

- “I don’t want to rock the boat”
- “I just want to go to work, do my job, and not get involved in the politics”
- “I don’t want to hurt or jeopardize my colleagues and their jobs”
- “If that’s what management wants to do, then who am I to say”
- “I agree with you, but don’t involve me in this”
- “I don’t want to be called a troublemaker”
- “I may lose my job”

# A Proposed Method to Minimize Occurrences of PDD

A. There should be an ethical/legal component to the requirements/design process in order to minimize occurrences of PDD, having the following basic elements:

1. Does the requirement contribute to the production of the simplest and best technical solution?
2. Does the requirement contribute to the production of the most cost effective technical solution?
3. Does the requirement and subsequent solution pass the smell test of Waste, Fraud and Abuse?

## A Proposed Method to Minimize Occurrences of PDD (Cont.)

**B.** As a method of enforcing the forgoing item, I suggest the following:

- 1.** At any point in the requirements or design phase of a program, empower any member to question the requirement or design.

This should be triaged, assigned and enforced in the same manner as a bug report.

- 2.** If there is agreement with management that the requirement/design needs revision to avoid PDD, then we do it and press on.

- 3.** If management disagrees, then a peer review committee should be convened which consists of only the technical staff, and specifically excludes management.

This committee then makes recommendations to management.

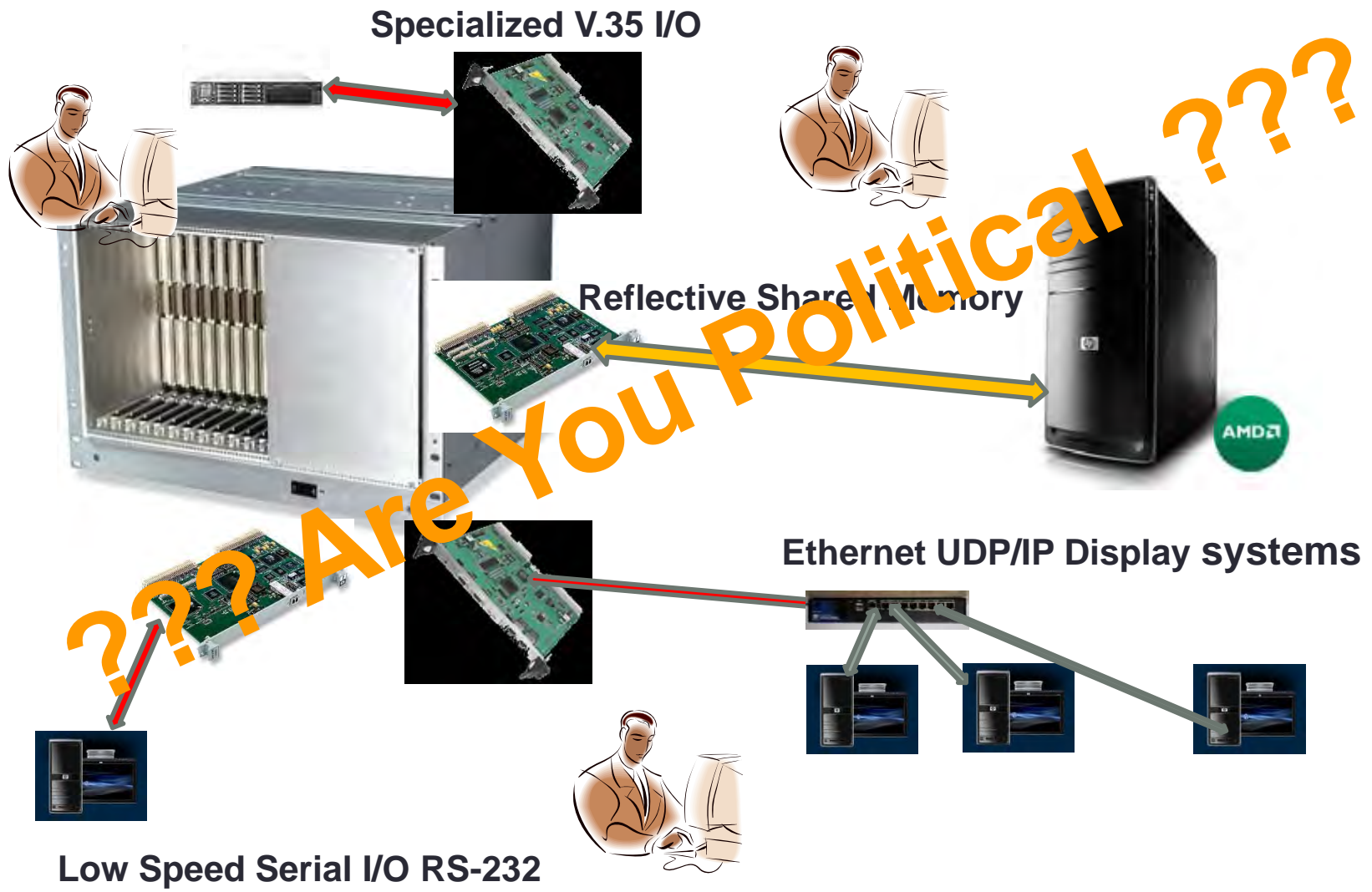
## A Proposed Method to Minimize Occurrences of PDD (Cont.)

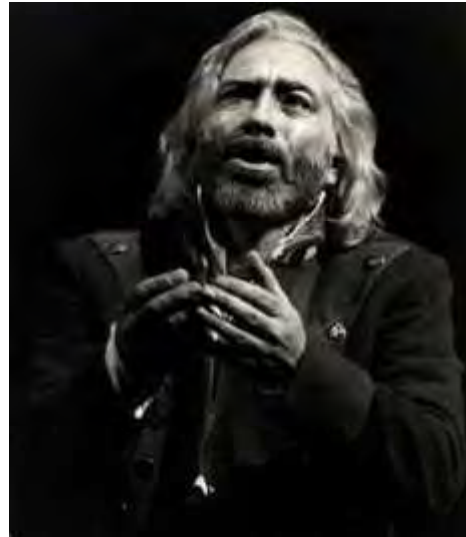
4. If management and the peer review disagree, then the issue is elevated to the customer/sponsor level. At this point the burden is placed on the owner of the perceived PDD requirement to justify on the basis of A:1-3 above.
5. It may be necessary at this point to elevate the issue to a formal and external agency (i.e. whistle blowing)
6. There needs to be accountability, with strong professional and legal penalties assessed against those found to be engaging in PDD
7. At all points, the person(s) questioning the requirement or design need to be protected from any penalty.

***People tend not to speak out when they see something is wrong. These procedures should be institutionalized in order to encourage a more proactive and responsible attitude among all stakeholders.***

# Conclusion

- As managers; if your organization has no Politically Driven System Development problem, I commend and congratulate you. You are being good “Civil **Servants**”.
- As managers; if you have PDD in your organization, shame on you, root it out. Step up to the plate and **earn the public trust** that has been placed on you.
- As technical staff members; do the right thing and speak out loudly and often against PDD.
- **In the commercial world of competition and profit/loss, Politically Driven Development would not survive the marketplace!**





Who am I?

If I speak, I am condemned.

If I stay silent, I am damned!

- Jean Valjean agonizing over a difficult moral decision in Les Misérables

# Acronyms

- I/O - Input/output
- NAVAIR - Naval Air Systems Command
- PDD - Politically Driven Development
- SSA - Software Support Activity