$25 Million in AI-Related Waste STOPPED*

Three Big AI Failures Yield Tough Lessons for Defense Electronics Firm

AI as Effective Code Generator

The SYMPTOMS:

- AI Performs *Brilliantly in the Lab*, *Fails Consistently* in Demos to U.S. Military
- At Six Months in, Find *NO HOPE of Coming Within 50% of On Time*, On Budget, As Promised
- $$ Millions Spent – No Contract Awards
- "SELL, SELL, SELL" + TECH, TECH, TECH + Low Bidder Culture = *Promises Made that Cannot Be Kept*

SPECIAL PROBLEMS in Defense Contracting

- Continual Changes, No Firm Requirements
- AGILE and Flexible, Rapid Development are preferred but *Defense Procurement Rules PREVENT*
- Government very good at creating competition, pushing work onto contractors (getting worse)
- Embedded Software Wins Create Overconfidence, Failures in More Complex Software

Big AI Lessons, How Use AI Well:

- Don't Allow High Level, Nebulous "AI Will Handle It" Work Packages
- Use Work Break Down Structure to Break AI Inputs / Outputs Down to Two Week Work Packages (User Stories)
- AI As Effective Code Generator: Improves Quality of Code, Cyber Security, Documentation, Testing and Integration at REDUCED COST
- Identified Where 45% of the Work Was Consistently Overlooked

Gave Management Data, Proof Needed to Say "NO" to Bad Demands by Customers, Sales

History of Code Generators: Not a Panacea but Can be VERY EFFECTIVE

Hard Lessons Learned Over Several Projects:

- **Electronic Border Wall**
- **Helicopter Flight Display**
- **Handheld Targeting System**
- **Advanced Vision for Reconnaissance Vehicle**
- **Helmet Mounted Display for F35**

Flawed strategy results in projects that might get sold, but will never produce a profit commensurate with capital, time and risk invested

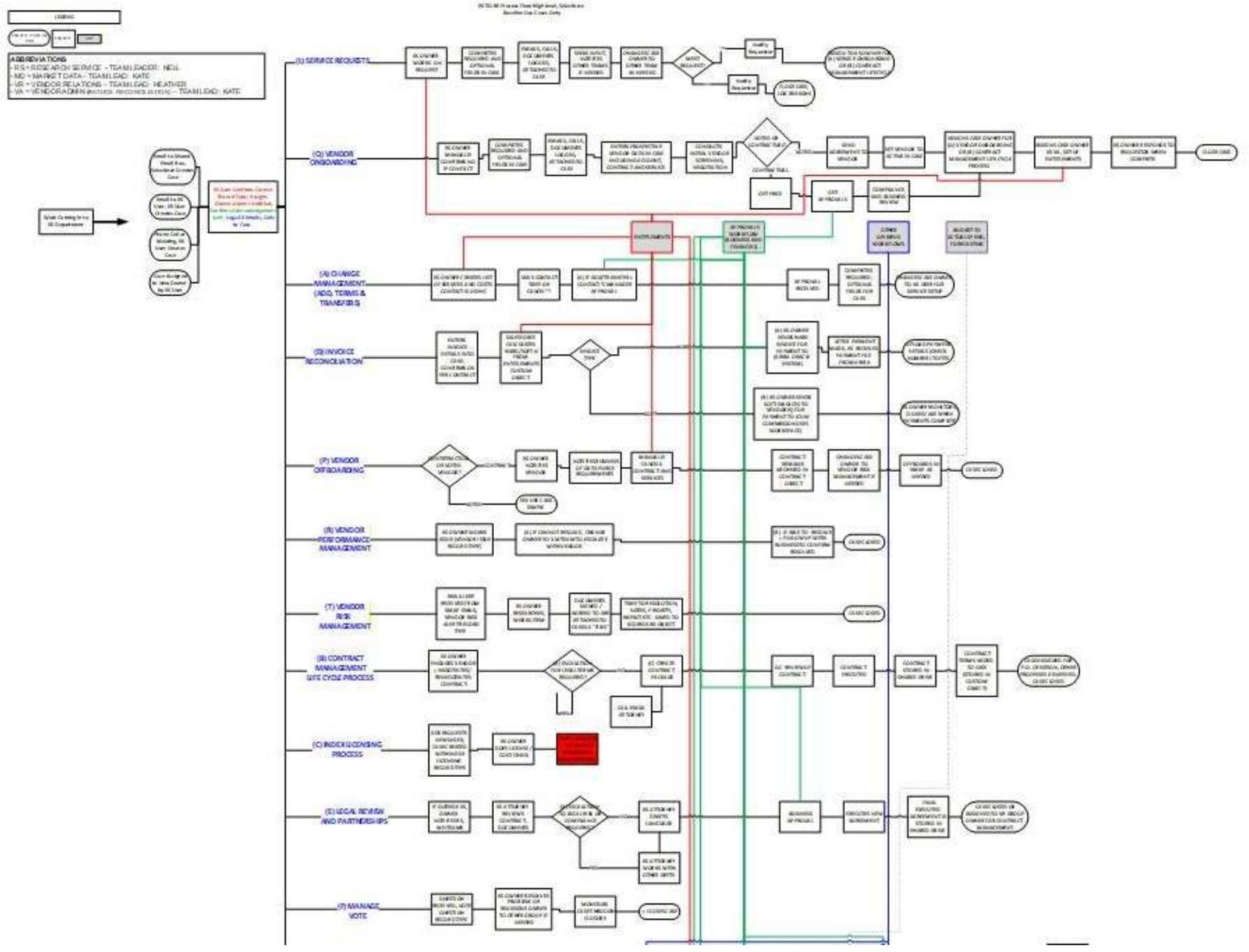
By Tom Ingram, PMP

NOTE: See disclosures* at bottom. This case is derived from actual projects but simplified to communicate key concepts to the non-technical, non-defense business person.

This case study shows, primarily in pictures, how years of tough lessons produced an AI breakthrough for a division of a defense electronics company. **The current U.S. software climate of "AGILE, go, go, go, code, code, code, deliver… demo… get paid" leaves many defects and much to be desired.**

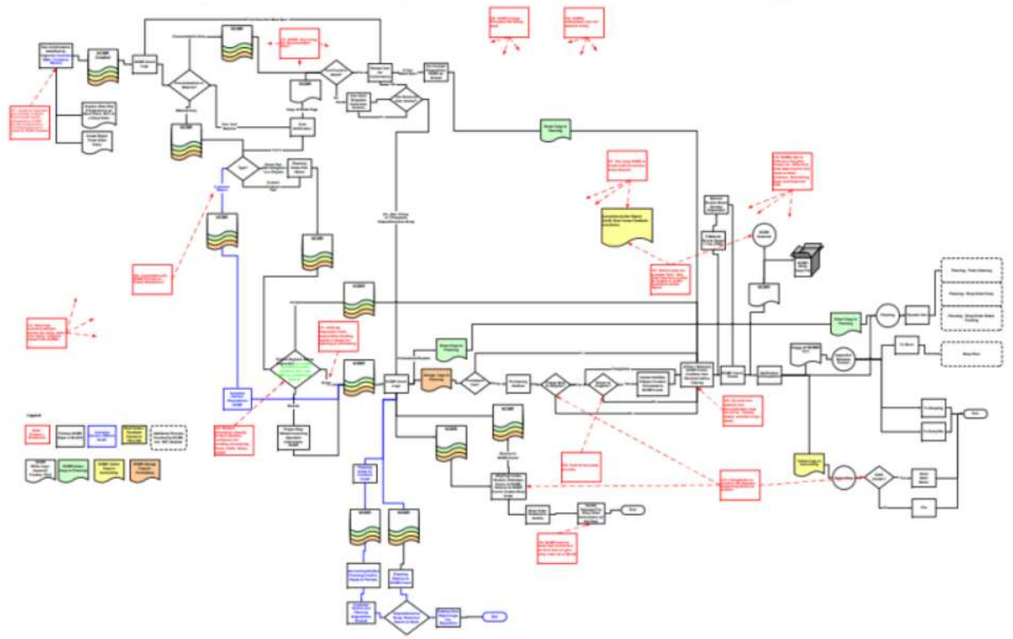
The tools outlined below (which have been around for decades and advocated by PMI) are often ignored. We present this case to show their value and how they were used to cut through hype, self interest and nonsense and produce a solid AI win from extremely difficult circumstances.

Use Cases – The Starting Point – Top 5 Use Cases Define 80% of the Value of the System

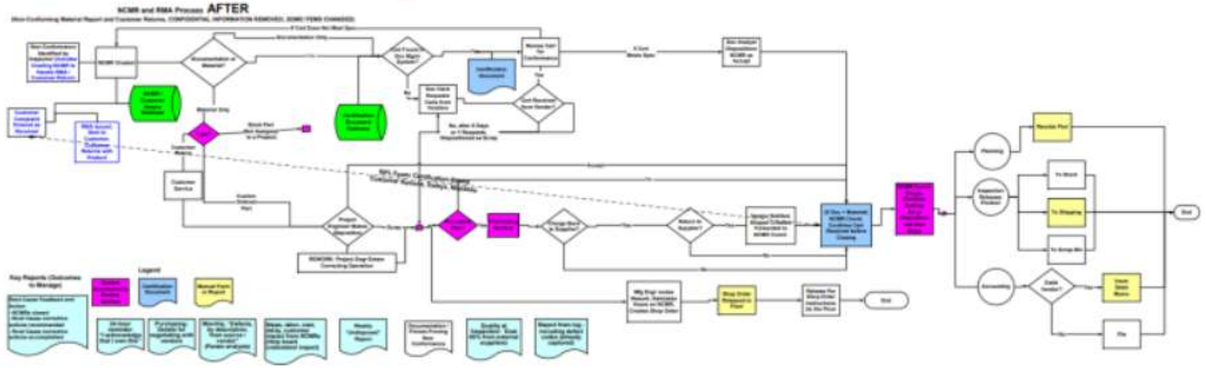


Fix the Process First!!! – The Single Most Important Lesson of 45 Years in Big Software

Before: 81 Step Process Complex Quality / Nonconforming Materials Manual Process



After: 81 Step Process Reduced to 46 Steps. Process Mapped and Workflow Software Prototype Developed within 60 days.*

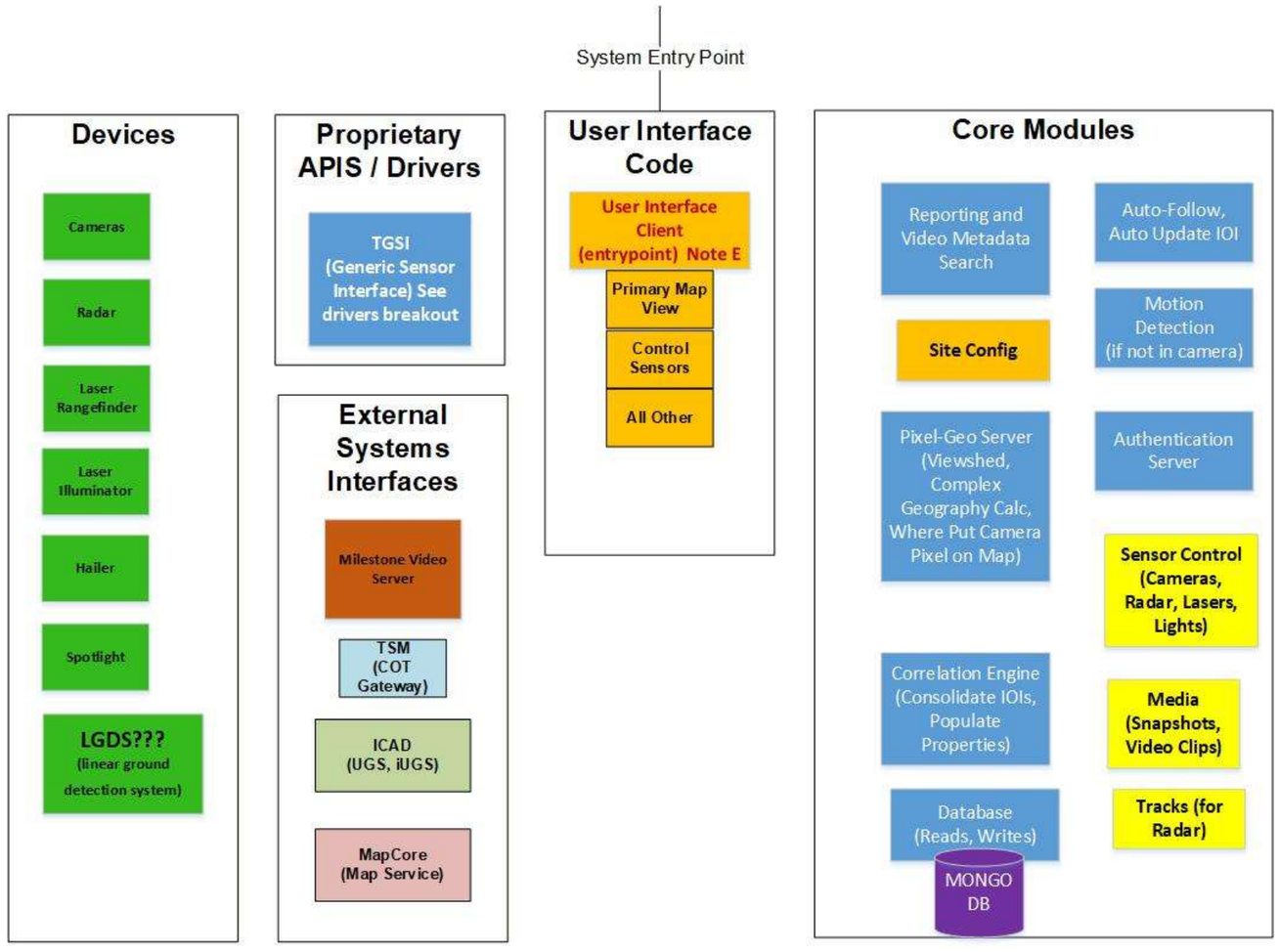


Requirements (Not pictured for brevity) CLICK HERE to view (password required)

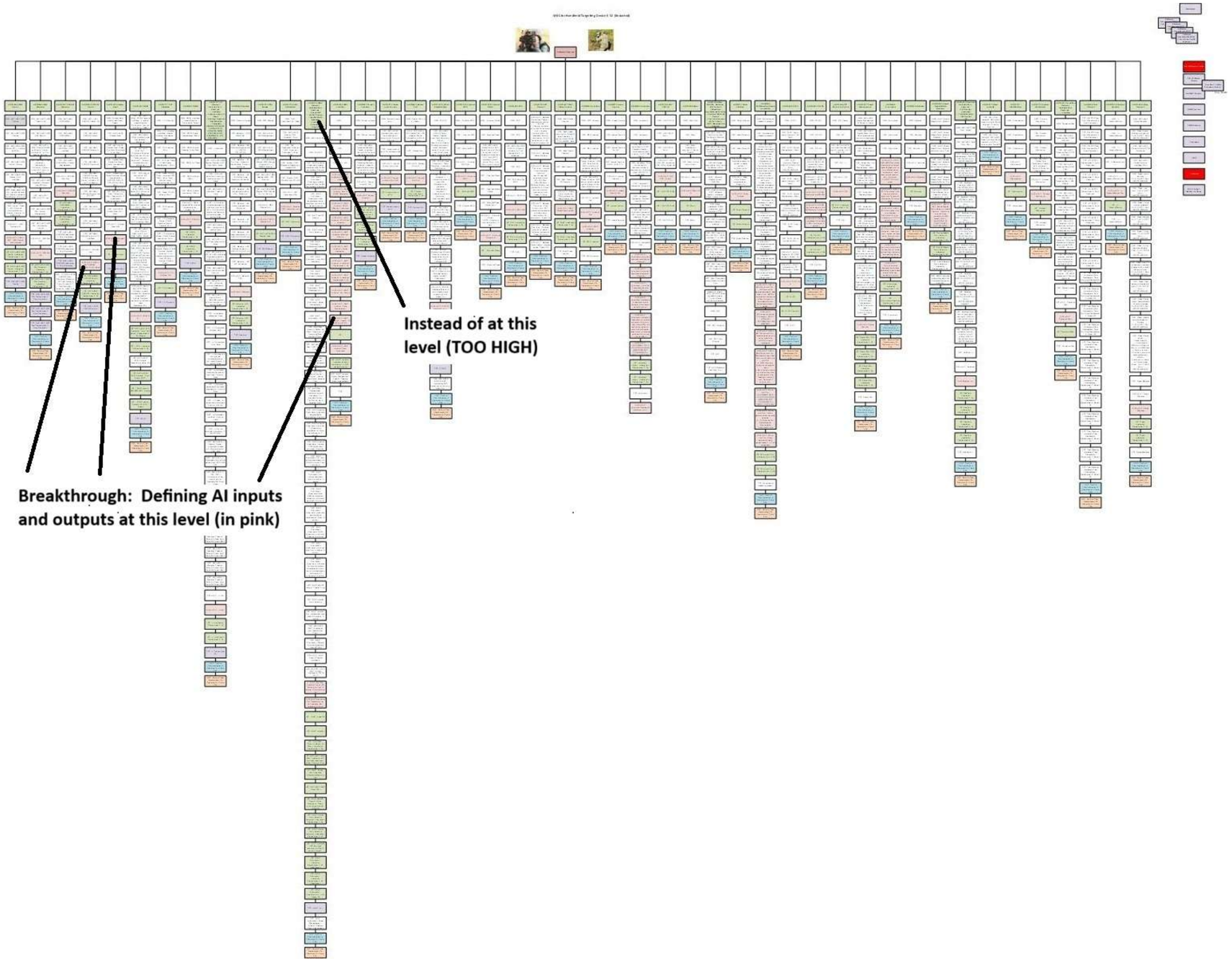
Requirements Triage – Reducing Scope to Highest Cost / Benefit (Not pictured for brevity) CLICK HERE to view (password required)

Prototype / Agile Kanban Board Alternative to Formal Requirements (Not pictured for brevity) CLICK HERE to view (password required)

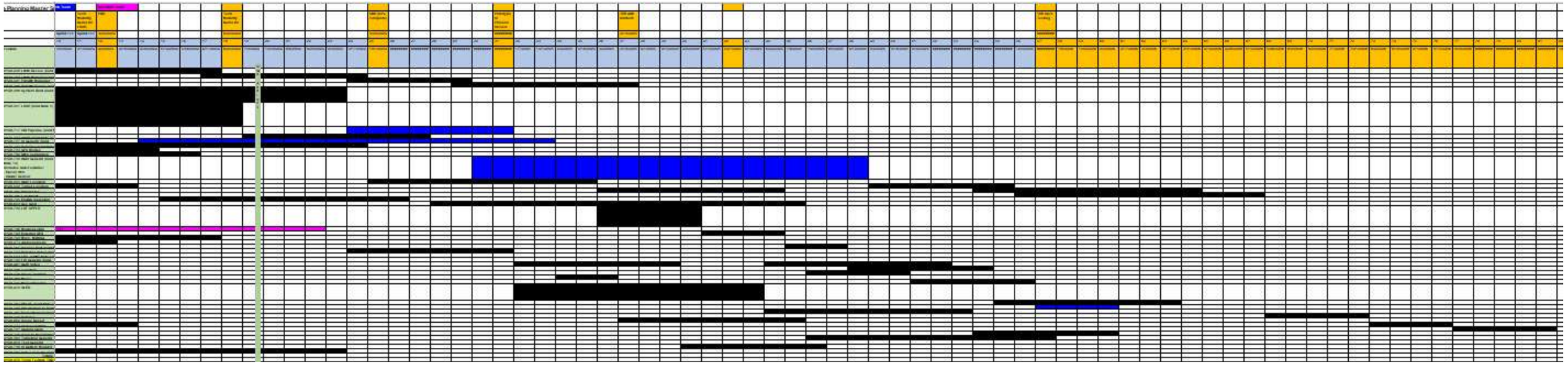
Architecture - Simplified



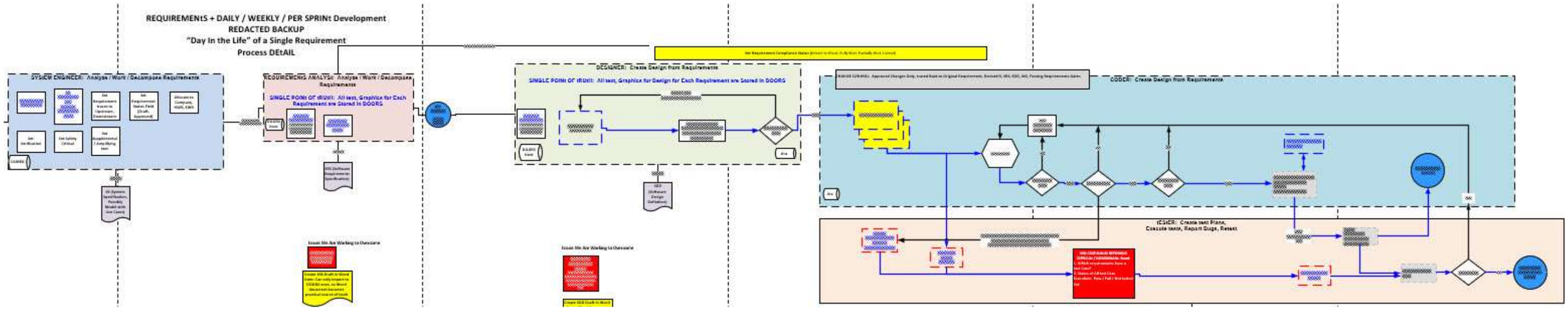
Work Breakdown Structure – JUST FOR SOFTWARE



Gantt Chart View of Work – shows 45% of work not planned for (blue and pink)



Development Processes / Disciplines (Agile – Waterfall Hybrid)



AI As Effective Code Generator:

Lessons learned from U.S. Defense Department: Using the WBS (Work Breakdown Structure) approach to define AI's inputs and outputs at the right level produces additional critical benefits:

1. **Provides an AUDIT TRAIL showing where the code originated and that it came from an AUTHORIZED CODE GENERATOR, approved by the D.O.D.**

2. **The authorization includes CYBER SECURITY RULES COMPLIANCE.**

3. **The code generators produce HIGH QUALITY DOCUMENTAION so code can be fixed and enhanced. The approach yields code that is:**

    a. **an UNDERSTANDABLE LOGICAL FLOW,**
    b. **defined in individual, TESTABLE CHUNKS**
    c. **HUMAN UNDERSTANDABLE DOCUMENTATION written into and alongside the code**
    d. **PREVENTS "RAMBO CODING" where a smart programmer quickly codes something only he/she understands, passes initial tests and moves on**

4. **The above approaches dramatically improve INTEGRATION TESTING and OVERALL TESTING**

The History of Code Generators yields several additional lessons: (CLICK HERE for article in process, password required)

- **Advocates have been claiming code generators will take over the programming business and put people out of work for decades. (I used to be one of them…)**

- **While useful for small tasks in controlled situations, code generators, including AI, are not capable of the higher level design, architecture and process aspects of large scale workflows involving dozens to thousands of users doing complex tasks.**

The Rest of the Story…

The difficult story of this division also demonstrates how **ROOT ISSUES can devastate the best efforts of those doing the work.** A summary follows. Contact us for more details on these complex topics.

- **Parent company problems**
    - Parent company leadership "control culture" **prevented multiple division CEOs from making necessary changes**
    - This division was forced to use the work product from an unreliable, self-centered peer division (instead of sourcing the work product on their own.)
    - Work done by these peer **units did not meet documentation, audit trail, security, testing and reliability requirements** of U.S. government. Millions of $$ wasted in failed efforts to rework bad products.
- **Strategy Mistakes**
    - **Barriers to Competition too low**
        - The government is very good at creating price competition
        - **90% of the time, competing on technical features alone fails to produce adequate margins** (Contact us for our research on this topic)
        - **90% of the time a "whole product solution" will produce superior margins and competitive barriers.** (Might include meaningful user input, software, hardware, process, integration, customization, training, implementation, cutover, break-fix and enhancements. Contact us for details.)
    - **Not solving a big enough problem to earn a strong margin and cover risk** (contact us on this complex topic)
    - **Lack of The Right Focus in Marketing / Sales Strategy:** A history of low bidder, commodity work resulted in **extreme pressure on both old leadership and the new CEO to sell higher value / higher margin work.** Implications:
        - **Chased Any Possible Dollar of Revenue.** The internal saying became "we are 6 inches deep and a mile wide".
        - Several acquisitions brought in new products and **new leaders with big egos**. These big egos had some success selling to early adopters in a smaller setting but **could not adapt to effective sales to pragmatic, mainstream government buyers**
        - A rogue business unit acquired too much power by making a few small, marginally profitable sales
        - **Layoffs provided the opportunity to remove four levels of technical leadership who were saying "NO"** to uncontrolled, unprofitable sales (complex – contact us for details)
- **Summary Examples**
    - **"SELL, SELL, SELL" + TECH, TECH, TECH + Low Bidder Culture = Promises Made that Cannot Be Kept**
    - **Ludicrous Underfunding:** The company spent several million dollars (IRAD – unreimbursed) on three iterations of a program which could have resulted in $50 million of high margin revenue. **While continually changing / adding features and scope, the customer and senior sales executives demanded due dates and an artificially low cost which could not possibly be met.** Examples include **45% of the work and cost simply removed** (as discussed above) and a **competent program manager replaced by a PART TIME program manager with NO EXPERIENCE IN SOFTWARE**
- **Some Positives**
    - New CEO changed mantra from "we will double sales in 5 years" to "6-7% profitable annual growth per year."
    - Parent Company – overall – maintains 25-26% gross margins, second highest in its segment (even with extreme struggles in this division)

*** Our Disclaimers and Cautions:** Please read the notes and cautions in this case carefully and contact us (tom@tomingraminc.com) for details prior to any decision to purchase or invest in a project based on this case. This is a high-level summary of complex efforts over many years. It intentionally hides proprietary information and combines multiple projects to demonstrate clear lessons. It does not provide extensive detail or completeness. The goal is an overall understanding of the author's view of key lessons. We make no representation that a similar outcome will be achieved for others.