BAE SYSTEMS

How We Got Software Off the Critical Path

Systems & Software Technology Conference

19 June 2007

Grace Kuznia
Dan McDowell
Brian Thomas

BAE Systems Land & Armaments
Armament Systems
Minneapolis, MN

Program Mistakes…Early Years

BAE SYSTEMS

- Minimal Modeling & Simulation (M&S)
- Long build schedules – over a year
- Top-down requirements flow
  - Late &/or unclear – enough blame for everyone
- Strict waterfall development
  - Nothing started until previous finished
  - Excessive unit testing – never "done"
- Massive integration effort – code "over the wall"
- Hardware available, waiting for software

Being At End Of Engineering "Food Chain" Is NOT Fun!

Learning From Our Mistakes… Next Program

BAE SYSTEMS

- Rapid Development Methodology (RDM)
  - Spiral / evolutionary software development
  - Light-weight processes
- Shorter builds
- Software simulations of hardware
- Requirements “discovery” from bottoms-up
- This time: Integrated on hardware in nine months
- Software available, waiting for hardware

Now – No Longer At End Of “Food Chain”!

How Did We Arrive at RDM?

BAE SYSTEMS

- Re-examined our development lifecycle
- Started initiative to address product development
- Derived an “Executable Architecture” (EA) to run early code
  - Used test script to verify new functionality up-front (during development)
  - Integrated code developed for projects
    - Three round fire mission was used as pilot to work out issues
  - Used M&S to aid development
    - User interface screens
    - Hardware simulations to understand equipment motion sequence
- Documented development steps that eventually evolved into the Rapid Development Methodology (RDM)

What We Accomplished with RDM…

BAE SYSTEMS

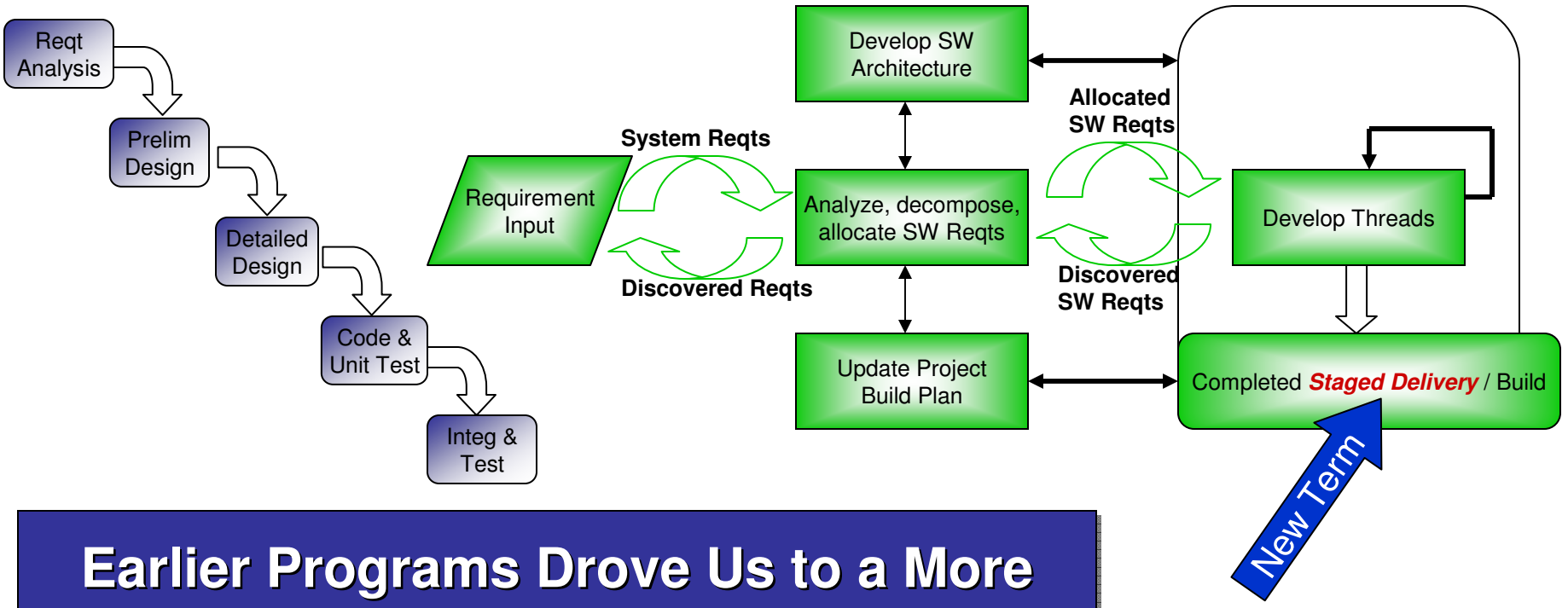
- Created software in short, iterative lifecycles
- Reduced reliance on late or incomplete requirements
- Integrated and tested software in less time (continuously-executable tactical software)
- Streamlined process activities
- Software available, ready for hardware
- Produced reliable, safety critical software components

RDM Definition:
An Iterative, Evolutionary Lifecycle That Handles the Uncertainty & Risk Inherent In Complex Software Projects

SW Development Approach

BAE SYSTEMS

- Before:
- After:



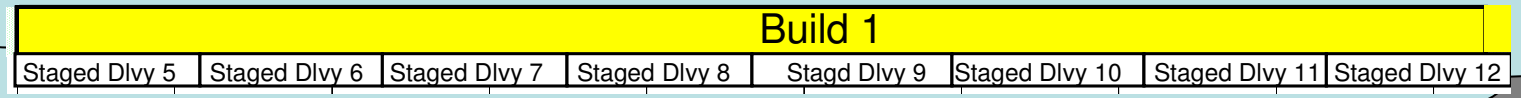
Earlier Programs Drove Us to a More Effective Approach

What Are Staged Deliveries?

BAE SYSTEMS

- Short, full software life-cycle development
  - called “Staged Deliveries” [reference: Steve McConnell’s Software Project Survival Guide]
- Each is approximately 75 working days long
  - Allows more effective management of schedules; always “complete” at the end of the 75 day period
  - Sixty five days of development; ten days for closeout, risk and planning
- Internal deliveries to ourselves
  - Allows for small victories along the way
- Avoids the “big bang” approach of the waterfall methodology
  - Ensures executable software throughout each staged delivery
  - Have tested functional code that builds upon previous Staged Deliveries
  - Enables us to mature software before formal delivery to customer
  - Allows us to readily adapt to changes (e.g., schedule, requirements issues, hardware availability, etc)

On our IMS schedules, the Staged Deliveries appear like this:



Technical Product Reviews

BAE SYSTEMS

- Focused on product development and software design not status
- Technical Review vs. Cost/Schedule
- Planned vs. Actual functionality implemented
- Twice per Staged Delivery
  - Mid-term
  - Final
- Always DEMONSTRATE compiled code running

Products Showcased –
Engineers' Achievements Recognized

A Few Words On Requirements….

BAE SYSTEMS

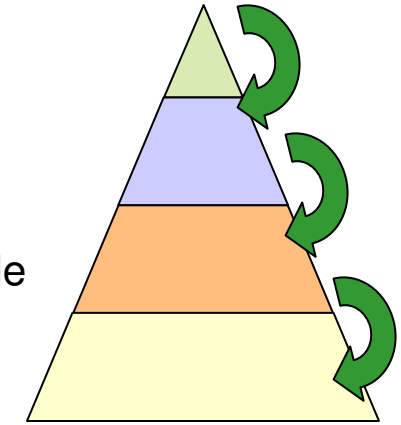
- Our going-in premise on requirements: You MUST figure out how to deal with flawed, late, missing, too much design in requirements
  - Otherwise hand wringing and hope becomes your method
- Reduced our reliance on late or incomplete requirements by…
  - Bottoms-up requirements analysis
    - Didn't wait – got started early
    - Proactively discover requirements based on domain knowledge
  - Continual requirements discovery
    - "Discovered" requirements then provided to Systems Engineering
    - Benefit: Minimized rework risk
  - Top-down requirements traceability
    - Performed traceability when system-level use cases available
    - Worked collaboratively with Systems Engineering to establish traceability
- Adapted to changing requirements during each iteration
- Focused on Thread development vs. CSCI development
  - Threads of functionality were developed and built/integrated
  - More frequent integration minimized difficult integration problems
  - Used a "build a little, test a little" philosophy

Collaborating With Systems Engineering Is *CRITICAL* To Successful Requirements Definition

Earned Value Techniques

BAE SYSTEMS

- Detailed schedules down to individual tasks for effective EVMS
- The “Drill Down” hierarchy….
  - Program Milestones
    - Customer Defined
  - Software Builds
    - Software Build Plan
  - Staged Deliveries
    - Integrated Master Schedule
    - MS Project
  - Individual tasks
    - MS Excel spreadsheet

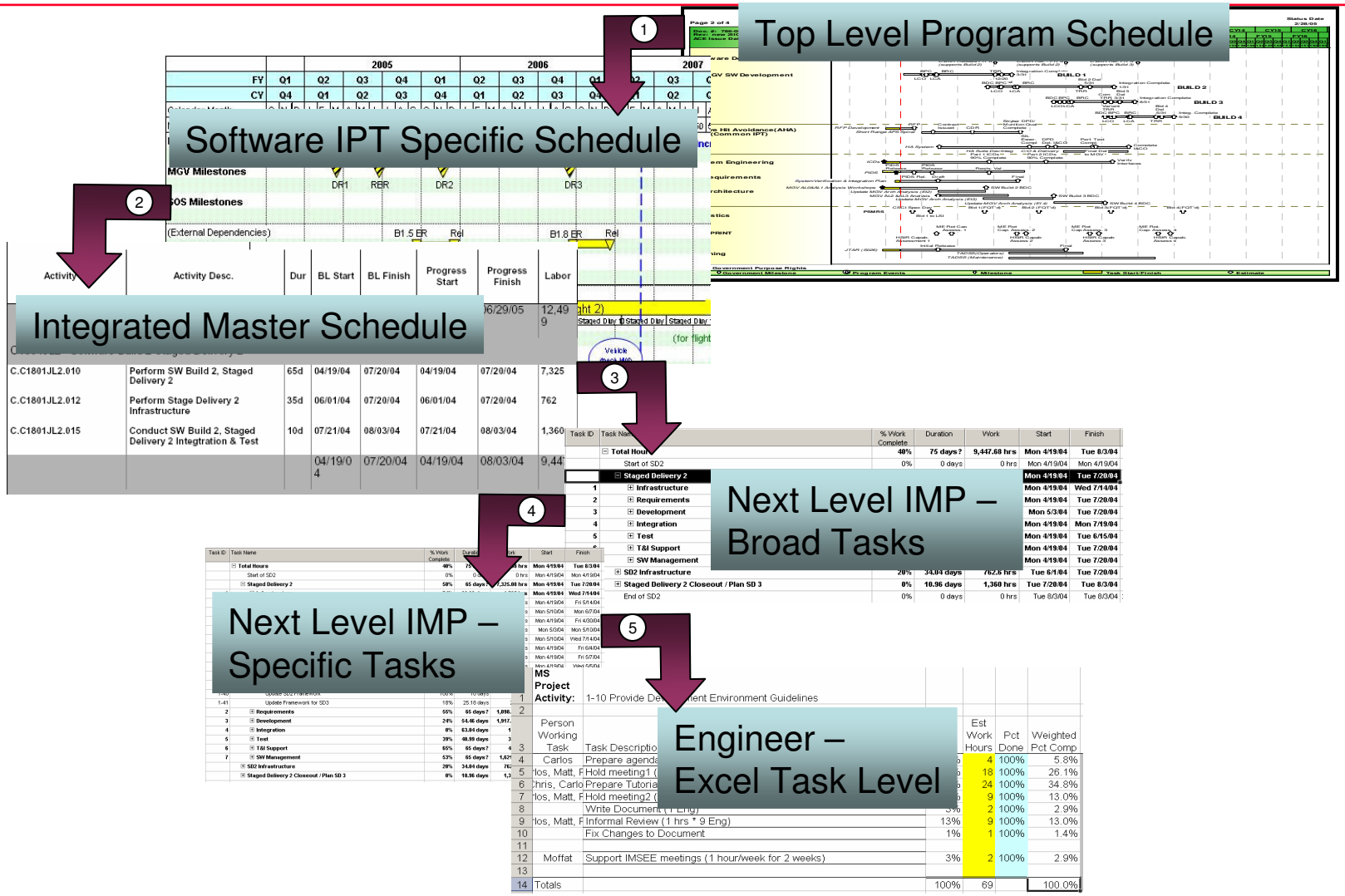


- Individual tasks no longer than 10 days
  - Tasks developed by each engineer
  - Track work performed by name

Lots of Work to Set Up – Worth It to Maintain

Schedule Traceability to Engineer Tasks

BAE SYSTEMS



A Few More Words About Staged Deliveries…

BAE SYSTEMS

- We discovered additional “drops” were needed within our 65-day iteration
  - Even 65 days was too long
  - Engineers were pushing their work to the end of the 65 days (student syndrome)
  - Between three and four drops is “about right” – every couple of weeks or so
    - Added additional informal drops throughout the staged delivery to spread out delivery of functionality
- Buffer Task (the ten days after the Staged Delivery)
  - Helps to clean up any remaining code or integration activities (risk mitigation)
  - Starts to focus engineers on the next Staged Delivery
  - Forces a planning (and thinking) period for the next Staged Delivery
  - Planned out just like activities within the Staged Delivery timeframe
- Appropriate customer visibility: Staged Deliveries
  - Takes a change in mindset from the customer
    - They have to be confident you have a plan and are executing to it even if there’s no detail in the IMS to tell them that
  - Very much a question of Trust
    - Good working relationship goes a long way to winning them over to the idea
    - We established that trust during an Integrated Baseline Review where we showed the same slides you saw earlier

Takeaways

BAE SYSTEMS

- Iterative vs. Waterfall
  - Test as we go
- Technical Reviews vs. Status
  - "Shows" products vs. Powerpoint
- Shorter Lifecycles
  - Demonstrate progress with executing code
- EV Planning and Tracking
  - Measure everything down to lowest level
- Requirements Philosophy
  - Use what you know; don't wait – seek out/understand customer needs

Other Things That Worked Well

BAE SYSTEMS

- Software Architecture
  - Layered architecture was developed previously, and continues to live on
  - Architecture patterns provide common infrastructure throughout the system
- Weekly Earned Value
  - Subtask schedules have tasks no longer than 10 days
  - Tasks are either done or not done (i.e., 0% complete or 100% complete)

Our Software Approach

BAE SYSTEMS

- Key tenets
  - Rapid Development Methodology
  - Build a little; Test a little
  - Staged Deliveries
  - Close collaboration with Systems Engineering
  - Lightweight processes
  - Weekly Earned Value

These Proven Practices Have Worked for Us

Obstacles Along the Way

BAE SYSTEMS

- Required cultural change on how people viewed their work
  - It's "Okay" to throw away code
  - Don't need to have ALL requirements before beginning SW work
  - Not everything needs to be 100% branch tested; Okay to find defects during integration
  - Hard to convince engineers that change is good AND can improve their work product

Summary

BAE SYSTEMS

- So what should you take away from our experiences?
  - Have fun, Work hard, Showcase your products!
  - Planning makes ALL the difference
  - Do something, anything – right away, even if you're not sure what
  - Try to remember a couple of our main points…
- We really don't have a "Magic Elixir", just a few techniques for us that have worked

Software Doesn't HAVE to be on the Critical Path – Even on Large Defense Programs!

One Final Thought…

BAE SYSTEMS

- If you're out front leading and no one is following, then all you're doing is taking a walk.

Or this…?

Thanks for your attention.
Good Luck…Have Fun!