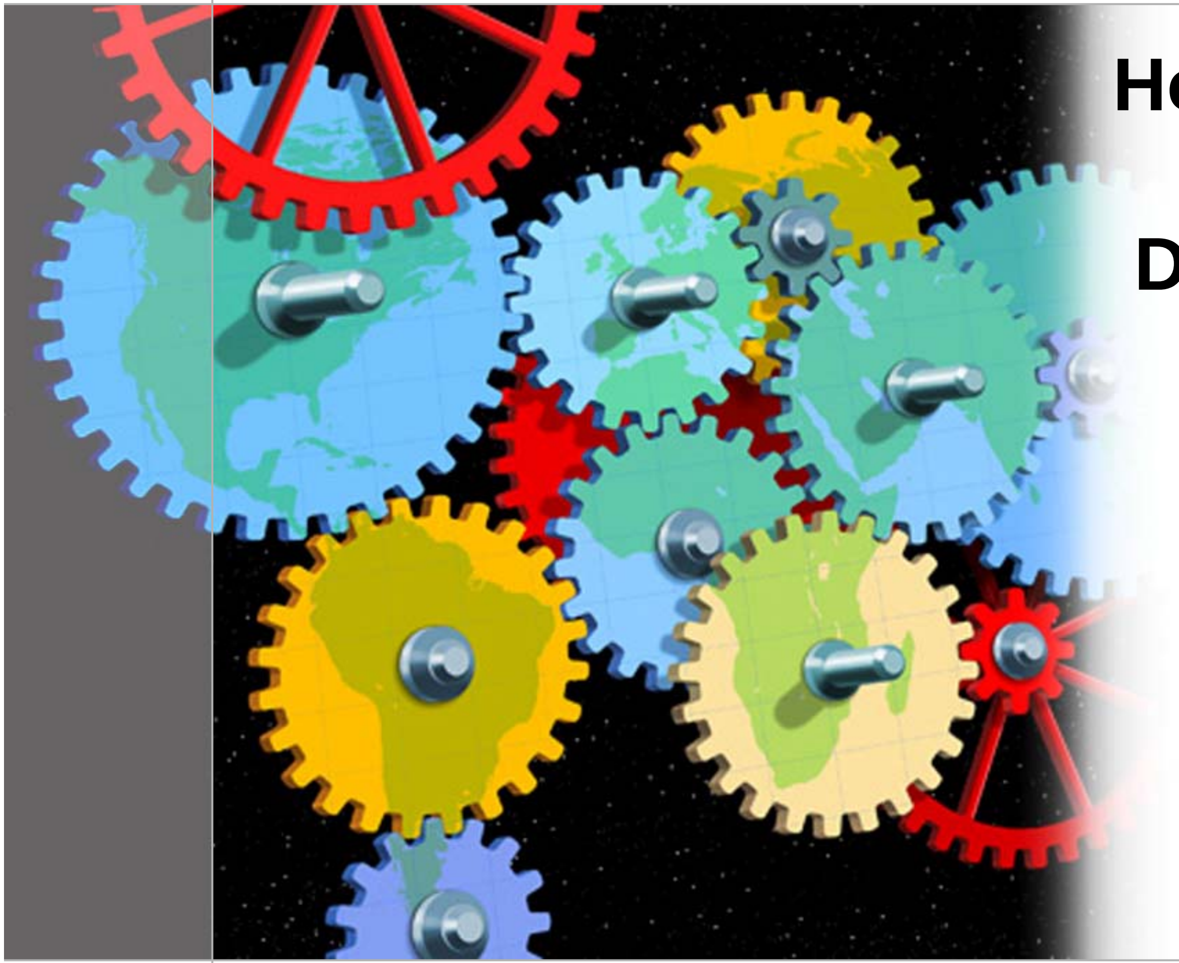
Raytheon

Customer Success Is Our Mission



How Raytheon Meets the Challenge of Developing Systems with Evolving Requirements

Linda Martz

Raytheon IIS Rocky Mountain
Engineering Director

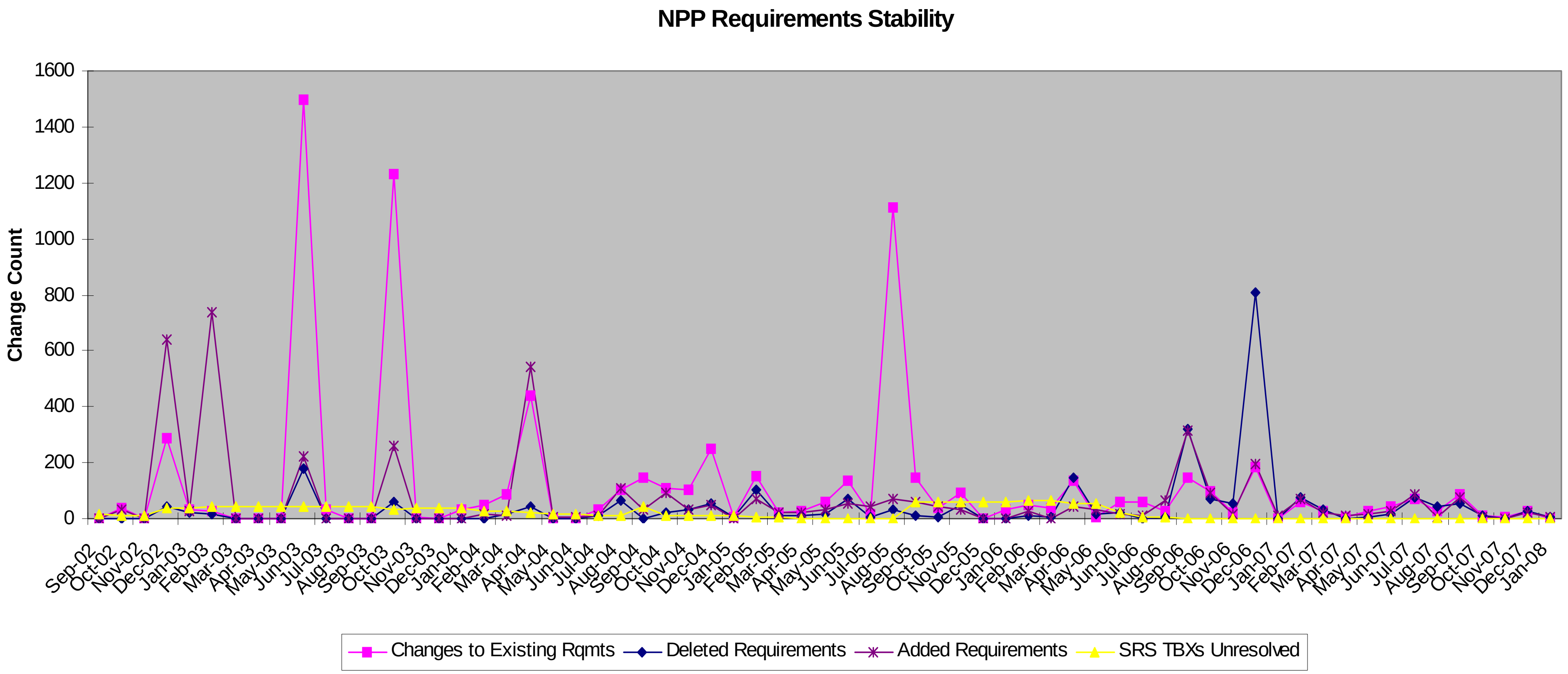
April 2, 2008

Challenges in Development and Acquisition Systems with Evolving Requirements

- Looking at the issues
  - Why the success of NPOESS Ground
    - CS3 completion of 1.8M LOC on schedule and budget, 75% reuse
    - IDPS through B1.5 on schedule and budget
    - Lessons learned from other successful SW development projects at Raytheon, Aurora using high levels of reuse

Why the success of NPOESS Ground?

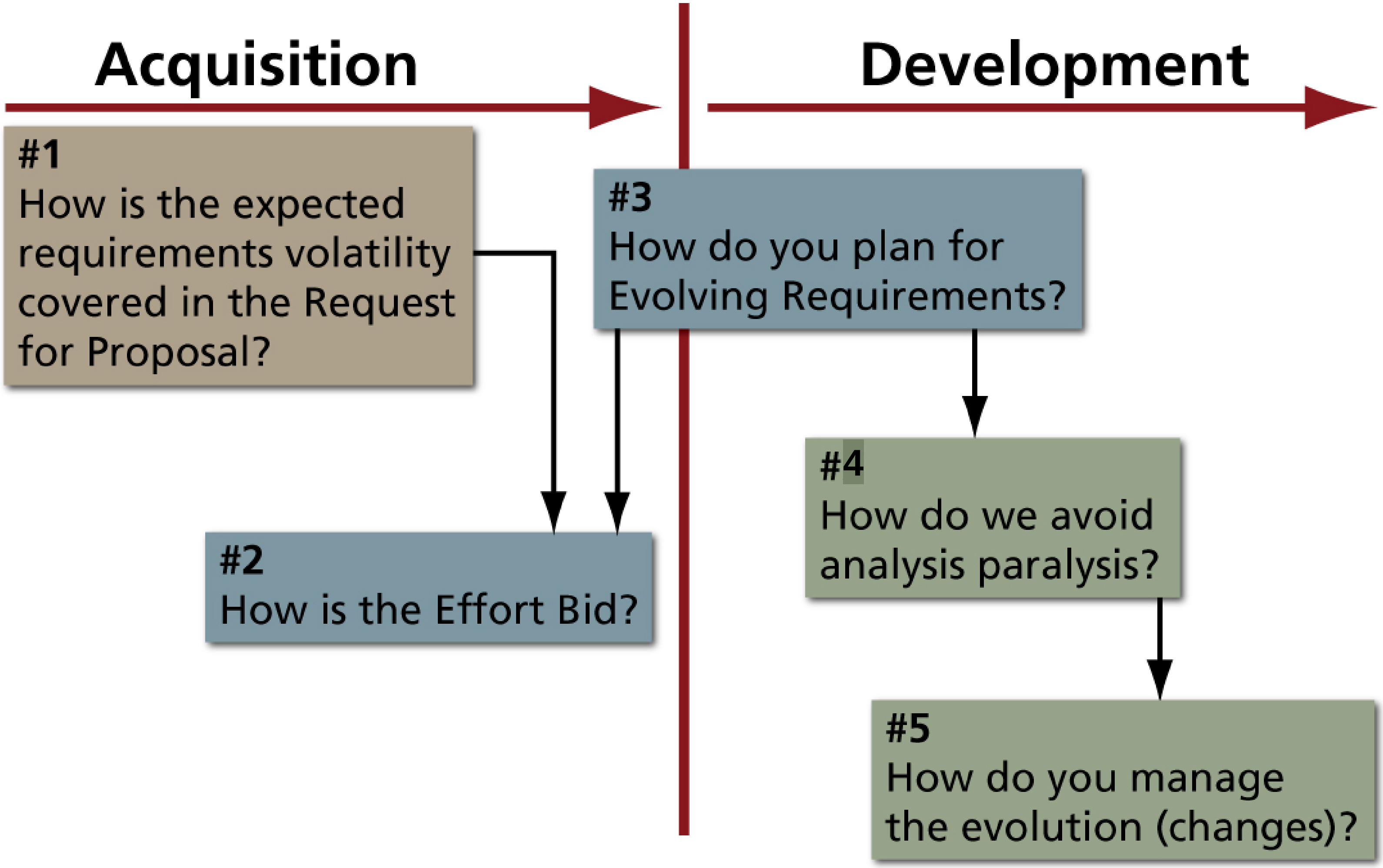
Development Metrics
NPP Subsystem Requirements



Total of 2016 C3S and 2942 IDPS NPP Subsystem Requirements

Requirement change peaks align with Iterative Builds

Challenges of Systems with Evolving Requirements


Acquisition
Development
#1
How is the expected requirements volatility covered in the Request for Proposal?
#2
How is the Effort Bid?
#3
How do you plan for Evolving Requirements?
#4
How do we avoid analysis paralysis?
#5
How do you manage the evolution (changes)?

Influences:

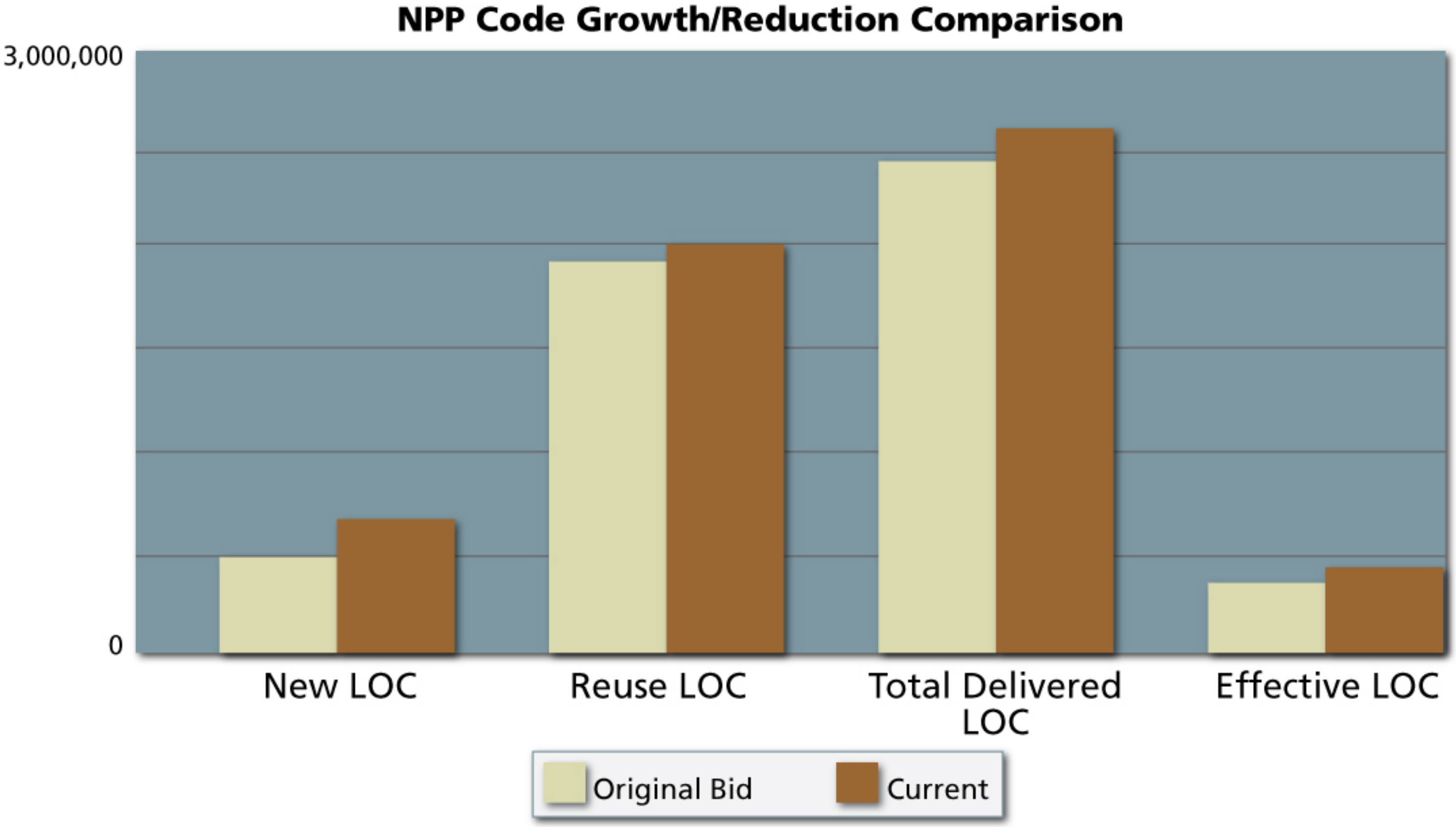
type of contract – ex: fixed price, cost plus
type of work – ex: R&D, manufacturing,

Contractor motivation – ex: sales vs. profit

Challenge #1 – How is the expected requirements volatility covered in the RFP?

- A baseline must be defined even if it's going to change
  - Baseline = technical + cost base + schedule
- It would be helpful, for the contractors, to have indications of where change is expected to occur
- A budget set aside for expected change should be clearly identified as whether it is to be included in the contractor budget or held by the acquisition organization
  - Make it not an easy target for stripping

Example: One metric that is effective by Evolving Requirements – Code Growth

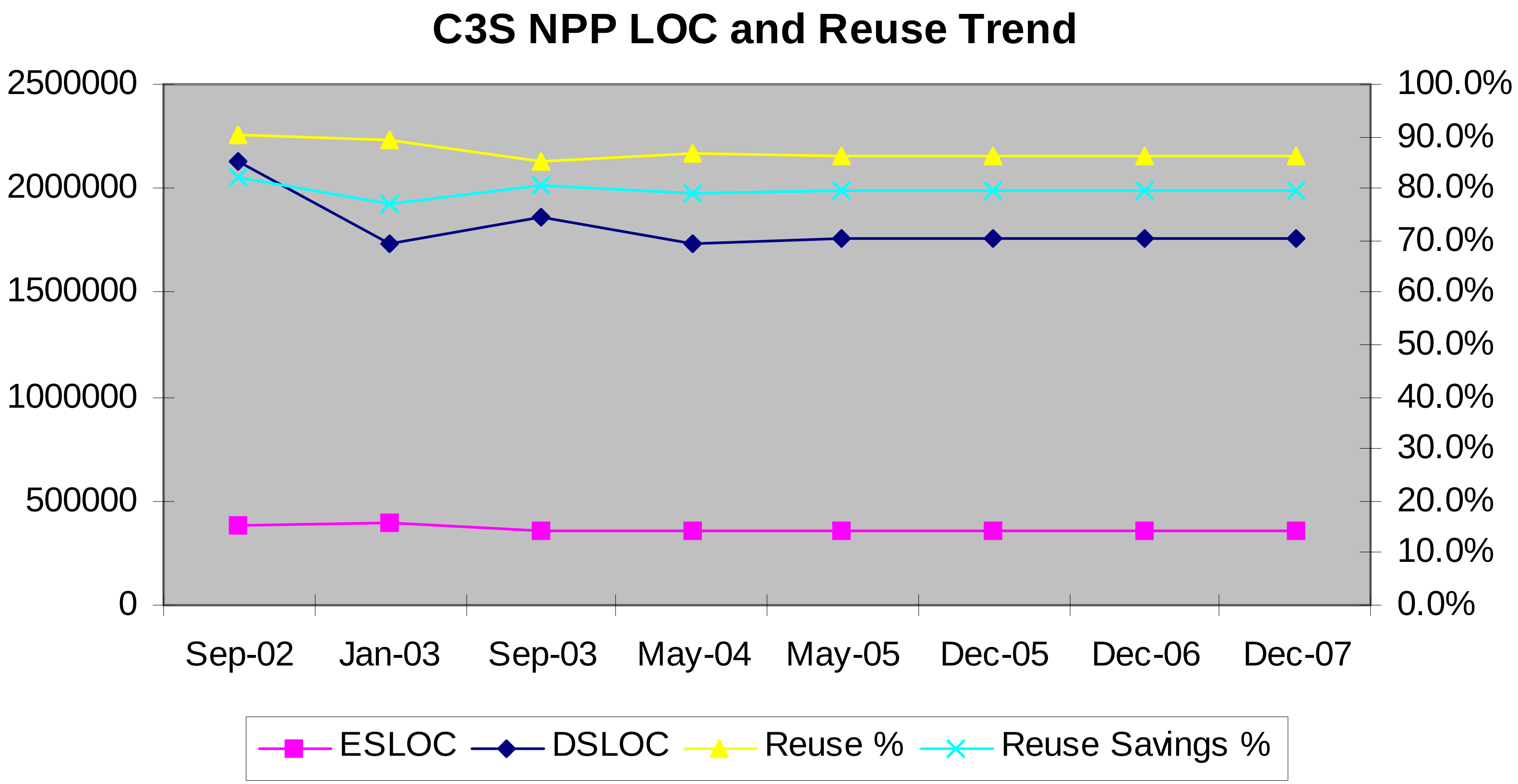


Government metric today has only one number for both types of code growth – in baseline Engr discovery (contractor obligation to manage) and baseline additions (government obligation). But RFPs ask to contractors to bid ONLY the defined baseline, no ECP growth.

NPOESS NPP Code Growth was primarily additions to the baseline and auto-generated code

Development Success Metrics
C3S NPP SLOC Reuse Trend

Raytheon
Intelligence and
Information Systems

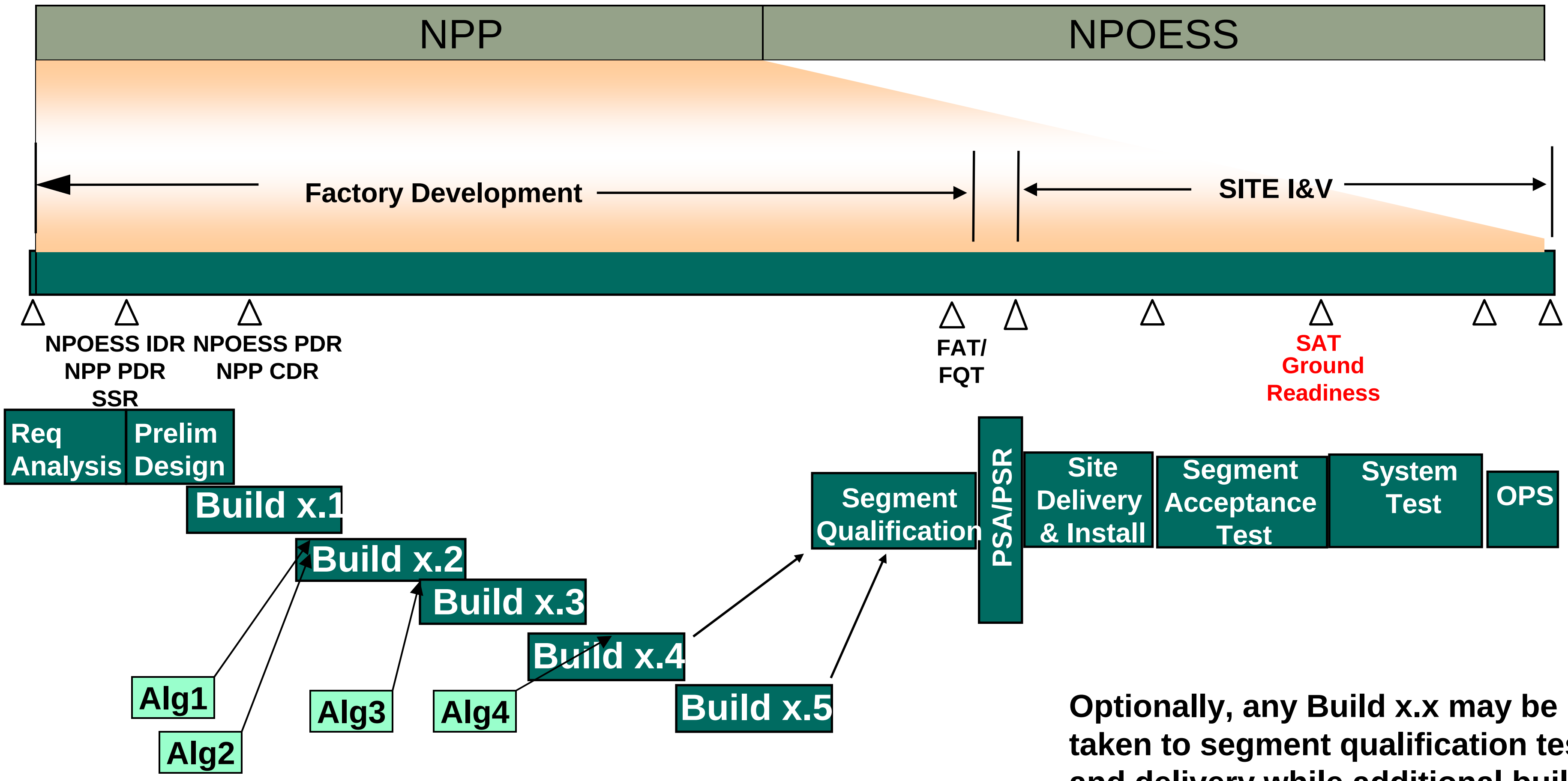


Challenge #2 –
How to bid the effort?

- Contractors MUST bid the baseline
  - – ACCURATELY and with HIGH CONFIDENCE
- Risk $ can be reserved for identified area of expected volatility
  - – For Software development – 3 ways to bid risk
    - Increased LOC
    - Lower Productivity
    - Separate identified risk pool
- Task Order/Management Reserve pools

Challenge #3 – How do you plan for Evolving Requirements?

Raytheon
Intelligence and
Information Systems



Algorithms follow a unique lifecycle and any algorithms available for Segment Integration are included in the Build I&T

Optionally, any Build x.x may be taken to segment qualification test and delivery while additional build x.x's continue on

Schedules that directly represent our processes

- Steps in the process are reflected in schedule/Earned Value definition and monitoring
- Early iterations include prototyping, reuse absorb, COTS evaluation
- Prior to or at the start of each Build/Iteration include considerations for
  - Requirements adjustment
  - Architecture and COTS changes
  - Technology insertion
  - Future build impacts including labor hours, procurement $, requirement ripple

Challenge #4 –
How do we avoid analysis paralysis

- Architecture that is highly componentized for insertion of firm areas and ability to change

- Iterative life cycle (including prototyping) - gives developers a comfort zone that they won’t go to far off track

- Focus not on artifacts as the end – but the system solution
  - Too much detail that has little impact on requirements or architecture may be wasted effort

- Proven risk management approach

Quality of our architectures and our reuse

- Component-based and service-based architectures are ready for evolution
  - Low coupling and simple interfaces between components
  - For NPOESS C3S we were able to bring in reuse from 5 sources (Equinox, Eclipse, DCCS, Sterling, CPR) and integrate the components because of well-defined interfaces
  - Parameter-driven in many components
  - Multiple languages have not proven to be an issue – C, C++, FORTRAN, Java
- Operationally-proven and (as-needed) certified components
- Formal exchange mechanism to make lessons learned and best practices visible

Multi-level Risk Management with risk management budget

- Multiple levels of risk review based on impact potential
  - Peer Reviews - Risk to components or interfaces.   (bigger risks may be initially discovered at a peer review)
  - Regular status meetings – issues/concerns raised, may turn into risks
  - Technical reviews – Risks reviewed, mitigations discussed, issues/concerns reviewed, actions addressed
  - Schedule and cost reviews – Earned Value analyzed and addressed,
  - Weekly schedule progress review
  - Risk Review Boards at Segment IPT level, Program level

- Risk Management Budget is the incentive to the team to identify risks
  - They know help is available for mitigation activities, or if the risk is realized
  - RMB not available if the risk is not identified

Challenge #5 – How do you manage the Evolution (changes)?

- As the Rolling Stones say, “You can’t always get what you WANT. You get what you need.”   (good vs. perfection)
  - Don’t confuse Out of Control Requirements with Mission Understanding and Happy Users
  - Ability to control scope, schedule, and cost while satisfying users is the TRUE ability to understand the mission of both the end user and acquisition authority
- Change is inevitable. Accept and manage it.
  - Change Control Board(s)
  - Risk Management Board(s)
  - Iterative Life-cycle, Requirements, ICDs, and Preliminary design baseline prior to first iteration, with change identified and impacted in following iterations.
- Use the power of requirements interpretation; trade offs to ensure system works and customer gets what they need in dynamic environment
  - Very often it is a large number of small scope changes that do the damage
  - Clearly defined pass/fail criteria generation during requirements generation
  - Each iteration reviews requirements to ensure user satisfaction

Multiple levels and implementation of Change Management

Raytheon
Intelligence and
Information Systems

- Multiple levels of change management based on impact potential
  - Code changes (reuse updates or identified deficiencies) through Software change board
  - Requirements changes through IPT review if no cost/schedule, eg. Grammar or terminology
  - Requirement changes through Program Change Control Board if cost or schedule baseline impact, within Program scope
  - Program and contract review if outside program scope

- Technical and programmatic change review and impact
  - Technical Baseline, e.g. Architecture and design documents, Test Cases/procedures
  - Process, e.g. Plans, work instructions
  - COTS – HW and SW
  - Reuse and new SW baseline
  - Contractual baseline

- Iterative-Incremental lifecycle uses each iteration as a change control mechanism (approval by CCB may be necessary to complete Start of Iteration Review
  - Review of all baseline changes from previous iterations
  - Include potential schedule updates
  - Risks and mitigations

- When and where matters
  - When is the change coming?
  - Where is it impacting the system?

Bottom Line

If both Acquisition team and Contractor team know the bus is coming, we can take steps to get out of the way!