Presented at the 2016 ICEAA Professional Development & Training Workshop -
www.iceaaonline.com/atlanta2016

Alternatives to SLOC in the Modern Environment

International Cost Estimating and Analysis Association

7–10 June 2016

Jeremy Goucher, Herren Associates

Dr. Jonathan Brown, Naval Surface Warfare Center, Dahlgren Division

NSWCDD/PN-16/200

Agenda

- Introduction
- Challenges to Software Cost Estimating
- Software Cost Estimating Process
- Status Quo—ESLOC
- Messages
- Automated Function Points
- Functionality
- Summary

Introduction

- **Individual software development projects for the Department of the Navy can be greater than $1B each**
  - **In a 2013 report, the Government Accountability Office stated "Information Technology (IT) projects too frequently incur cost overruns and schedule slippages …"**
  - **The same report stated that VistA-Foundations Modernization (FM), Global Combat Support Systems (GCSS)-Army, GCSS-Marine Corps, and the James Webb Space Telescope (JWST) reported a combined $1.3B in cost overruns largely due to poor cost controls**
- **Most modern development projects incorporate some software development requirements**
- **Most currently fielded technology systems require capability upgrades**
- **Will discuss three new methods for estimating software development cost, each suited for particular phases of the acquisition process**

Pre-milestone (MS) A	Automated function points (FPs)
MS B	Functionality
Post-MS C	Messages

Requirements Definition

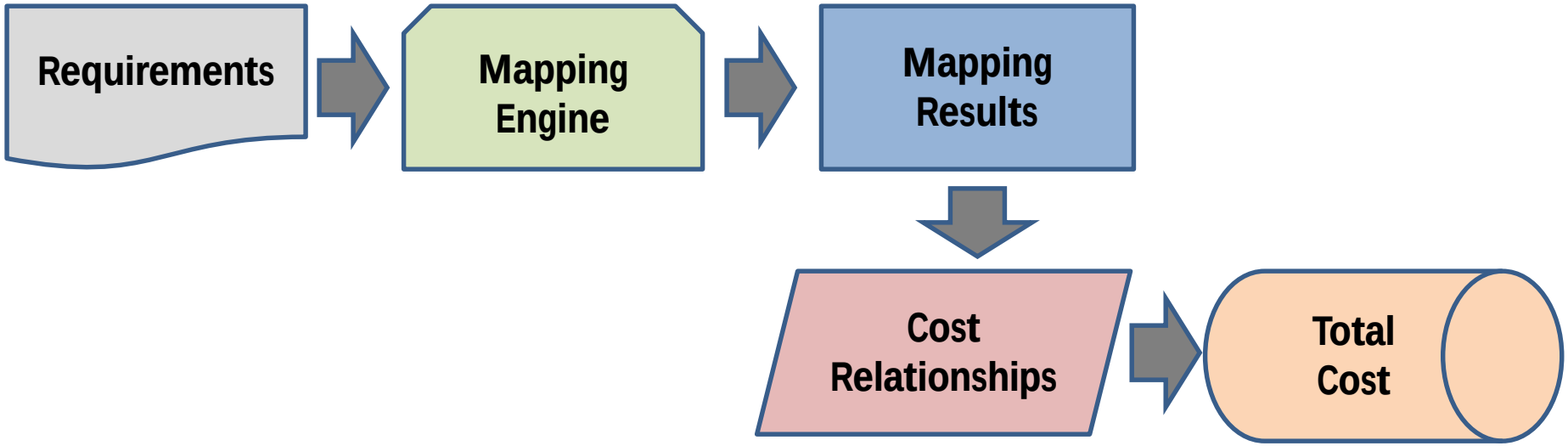
- **Requirements**
  - **Are any Function, Capability, etc. that is desired in the final product**
    - Examples
      - GUI
      - New Sensor
      - Increased Tracks
      - Increased GPS Accuracy
  - **Are not necessarily formal requirements documents as defined by DoD and other agencies**

Requirements are how you describe what your system will do!

Challenges to Software Estimating

- No physical characteristics
- Size metrics are proxy for labor requirements
- Analogous efforts are not necessarily predictive of future efforts
  - Varying maturities of software levels
  - Difficult to separate software growth from scope growth in cost reports
  - Varying degrees of effort required for modified and reused SLOC
  - SLOC counts can vary +/– 50 % based on the tool used

Typical Software Cost Estimating Process



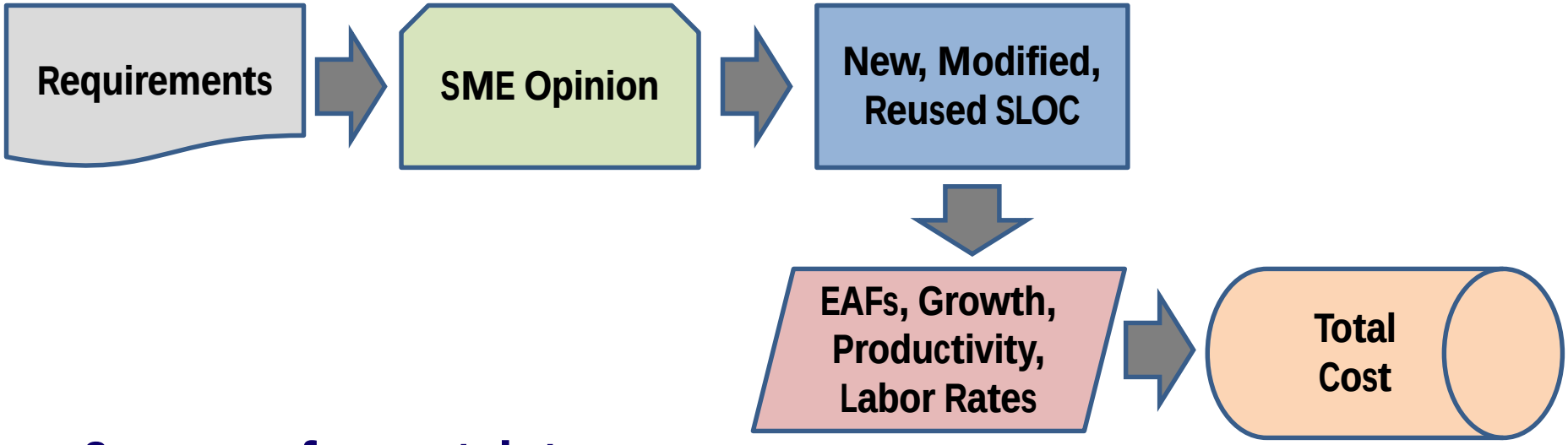
- Requirements are defined by stakeholder needs
- Requirements are mapped to something that can be related to cost, typically software size
- Cost relationships are applied
- Process results in cost estimate

STATUS QUO—ESLOC

Typical SLOC Based Estimating

- Uses new, modified, and reused source lines of code (SLOC)
- New, modified, and reused effort adjustment factors (EAFs) are applied
- Software growth rates are applied
- Software labor productivity is applied
- Process results in total hours, which can be multiplied by labor rates to get total cost

SLOC Based Estimating Process



- **Sources of uncertainty**
  - SLOC counts
  - EAFs
  - Growth
  - Productivity
  - Labor rates

Subject-matter expert (SME) opinion engine contains unquantifiable risk not covered by software growth

SLOC Based Estimating Drawbacks

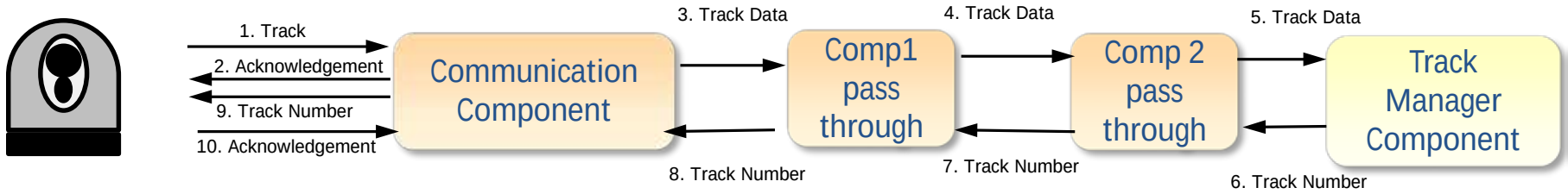
- New, modified, and reused SLOC are typically estimated by SMEs with analogous cross-checks
- Risk is applied to the SLOC quantities, EAFs, growth, productivity, and labor rates
- Risks are difficult for program manger (PM) to control
  - PMs have control over productivity, schedule, scope, and labor rates, but do not directly control EAFs or growth rates
- Difficult to compute joint confidence levels (JCLs) since calculation is not tied to schedule
- Does not fit the Agile framework well

MESSAGES

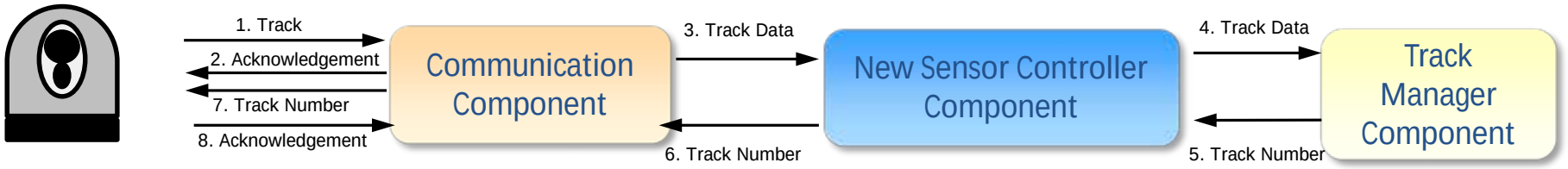
Number of Messages Within the Software

- Any communication between two elements of a system
- Part of the system architecture and likely well understood
- Messages are highly correlated with SLOC
- Automates and standardizes the method of counting SLOC
  - Commercial tools are available to count messages in an existing architecture
- Particularly powerful method for software upgrades of existing systems

Hypothetical Sensor Example



Example: CURRENT



Example: FUTURE

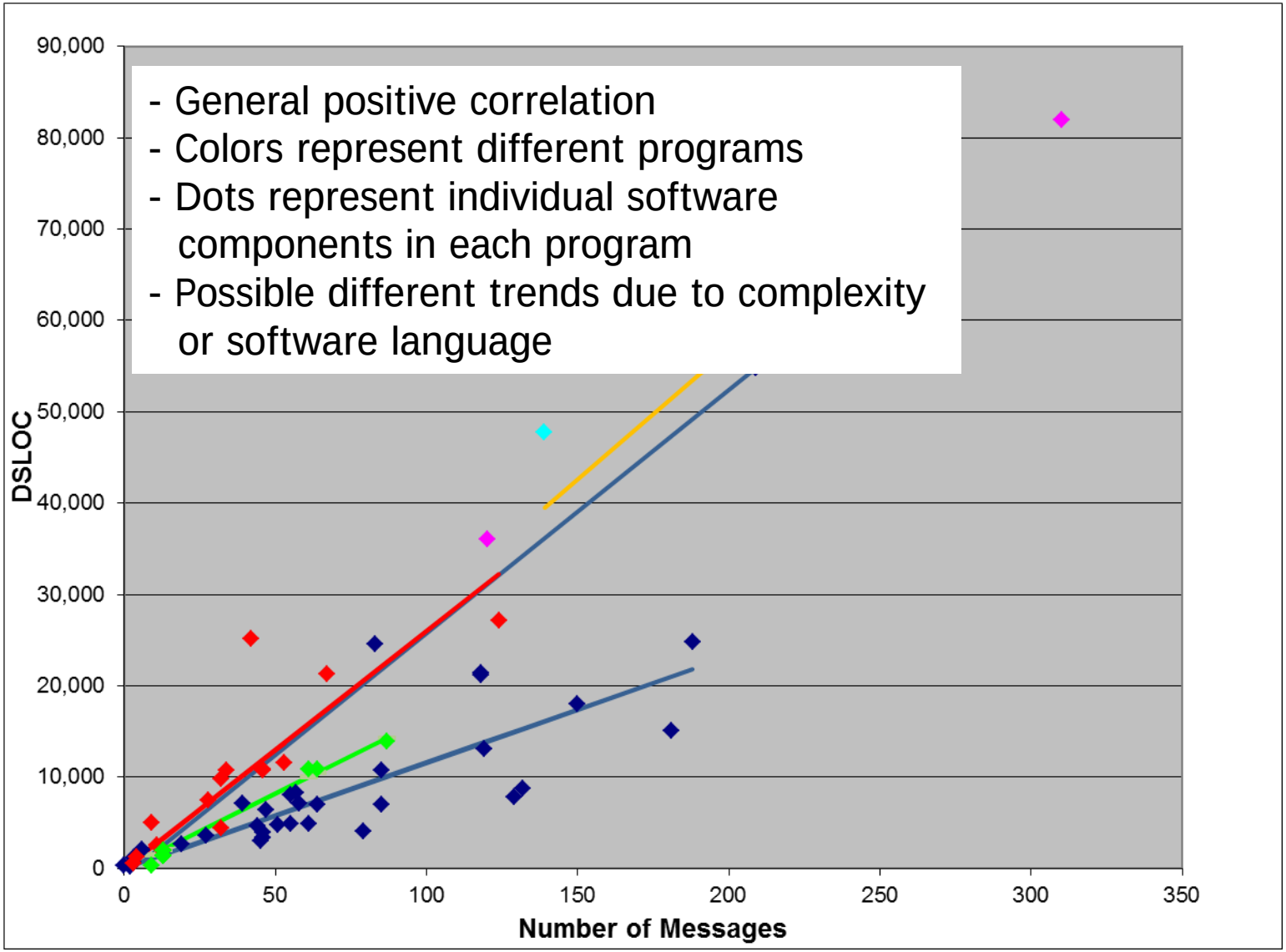
- Example Update: Add a sensor controller software component to reduce pass through and control hypothetical new sensor
- Adds four new/mod messages which replaces six old messages
- Keeps four reused messages

Messages (cont'd)

- Requirements mapped to the required number of messages
- Cost estimating relationships (CER) based on historical data translates messages to SLOC as shown in graph on next slide
  - Future research will result in $/message CER, eliminating the SLOC step
- Productivity is applied to SLOC, resulting in labor hours
- Labor rates applied to phased labor hours to get cost
- Phasing plans can be developed based on historical programs

Messages improve the mapping process

Message to DSLOC Correlation



New, Modified, and Reused Messages

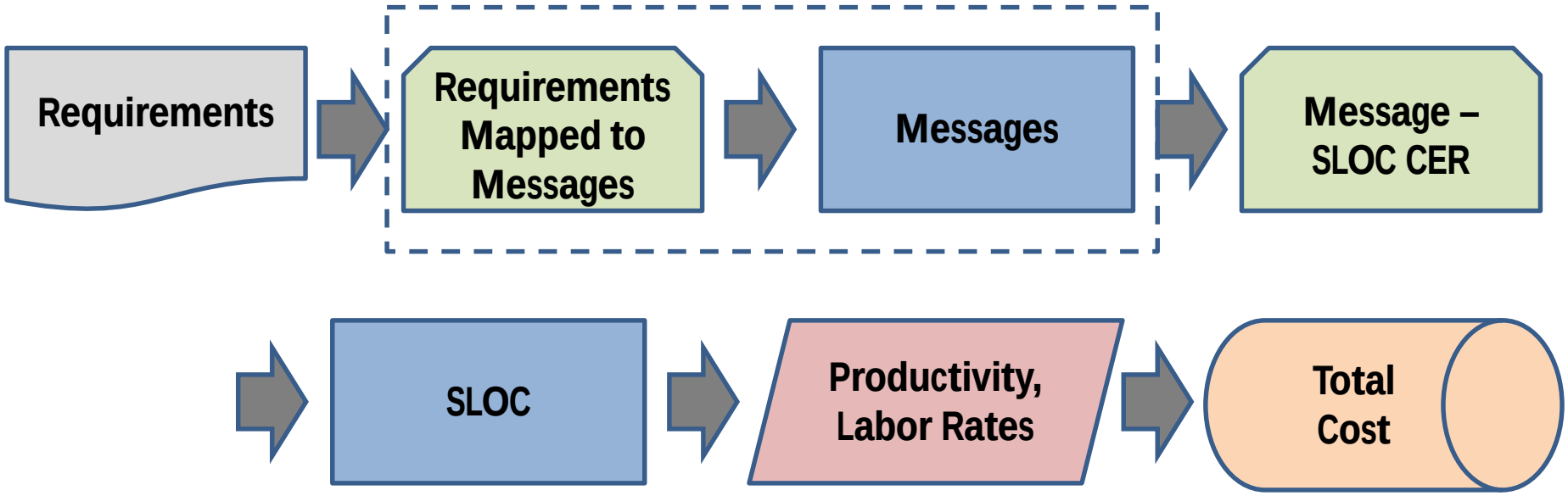
New Message
- All new SLOC
- Some new, modified SLOC
- Unlikely all modified
- Unlikely reused SLOC

Modified Message
- All new SLOC
- All modified
- Some new, modified, reused SLOC
- Unlikely all reused

Reused Message
- Necessarily all reused SLOC
- No new SLOC
- No modified SLOC

Effort levels can be tracked for messages the same as they are SLOC using the same reporting templates.

Messages (cont'd)



- **Sources of uncertainty**
  - CER coefficient
  - Productivity
  - Labor rates
  - Message count

Adds a mapping process; eliminates guesswork

AUTOMATED FUNCTION POINTS

Function Point Definition

- FPs are comprised of inputs, outputs, inquiries, internal data, and external data interfaces (Automated Function Points, 2014)
  - Independent of technology, platform, and software language
  - Manual, labor-intensive process
- Results vary +/– 35% based on various factors, including the project's complexity and analyst's skill level (Software Size and Productivity with Function Points, 2013)
- Drawbacks have prevented widespread Department of Defense usage

Automated FPs (AFPs)

- Recently, commercial standards were implemented to automate the FP counting process (Automated Function Points, 2014)
- Process digitally analyzes source code to produce a resident FP count
  - Similar to Unified Code Counter (UCC) or other tool counting LOC
- Reduces counting time
- Improves consistency using a defined algorithm
- Requires mapping mechanism between requirements and FPs

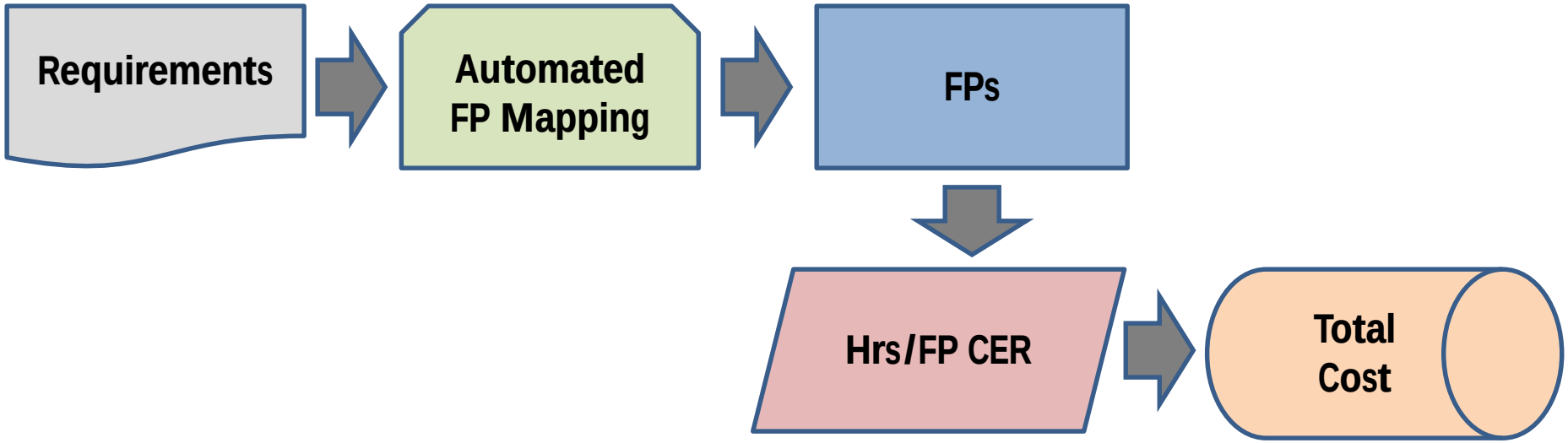
An AFP counter makes it time-feasible to analyze historical programs and develop historical cost, hour, and FP count databases

AFPs (cont'd)

- To fully realize benefits, it's necessary to improve mapping engine
  - Requirements mapped to FPs
- Nature of FP more easily lends itself to algorithmic mapping
  - For example, feature location (Small Business Innovation Research)
  - Algorithmic mapping would have known, quantifiable, repeatable uncertainty that could easily be modeled
- CER developed to estimate cost/FP
  - Developed using FP database
- Risk modeled using prediction intervals from CER regression
- FPs can roll up to functionality, which rolls up to builds, which can be mapped to schedules, resulting in JCL

An AFP, when combined with an algorithmic requirements process and historical CER, quantifies the uncertainty measurement and speeds up software size estimate development

AFPs (cont'd)



- **Sources of uncertainty**
  - Hrs/FP CER coefficient
  - Automated FP mapping

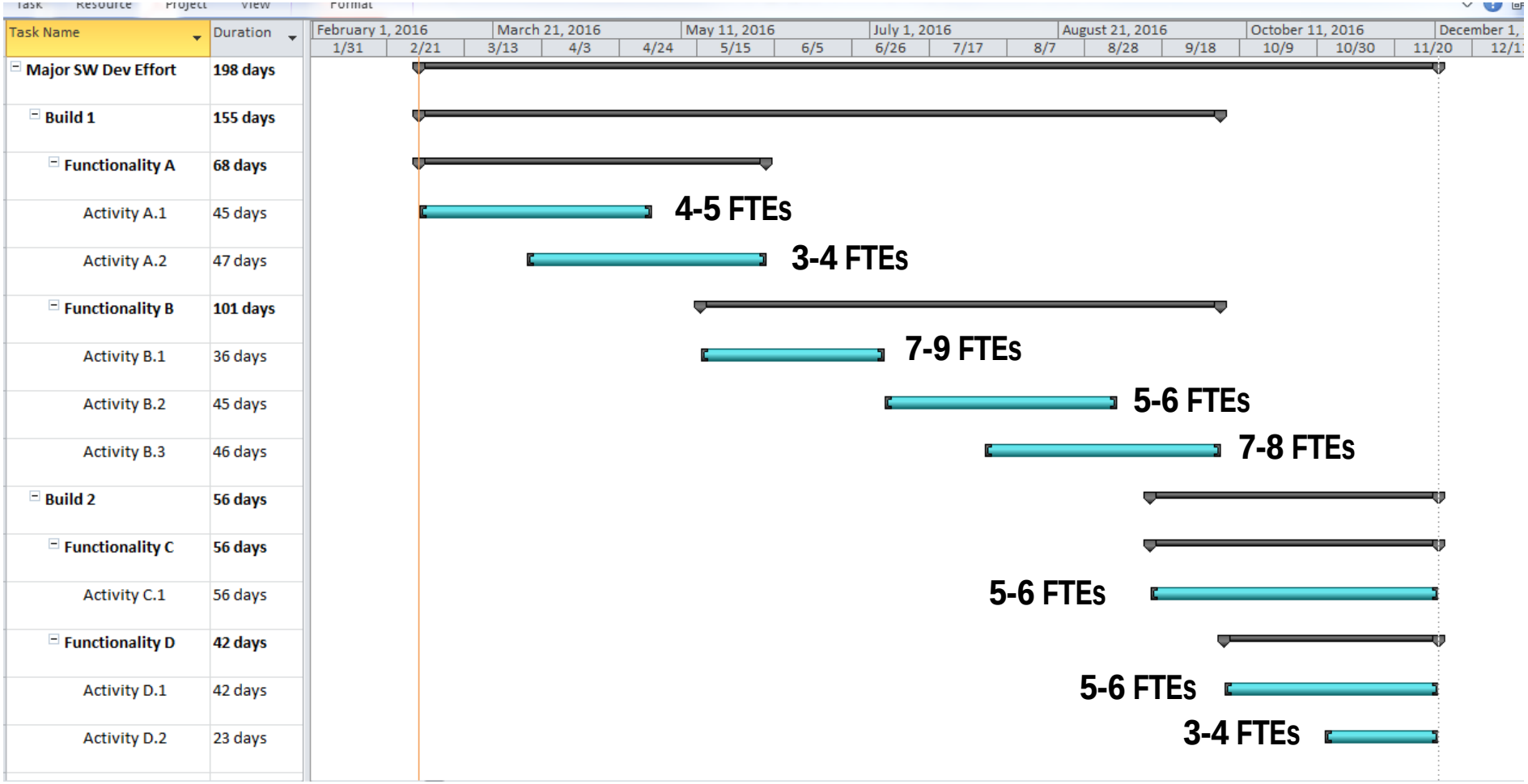
Simplifies and quantifies risk
Eliminates/reduces SME inputs

FUNCTIONALITY

Functionality

- Easily mapped to requirements
- Easily mapped to build plan/schedule
- Suitable for Agile products
- Requires SME to estimate full-time equivalent (FTE) requirements
- Simplifies risk calculation, suitable for JCL
- Risk more manageable by PMO

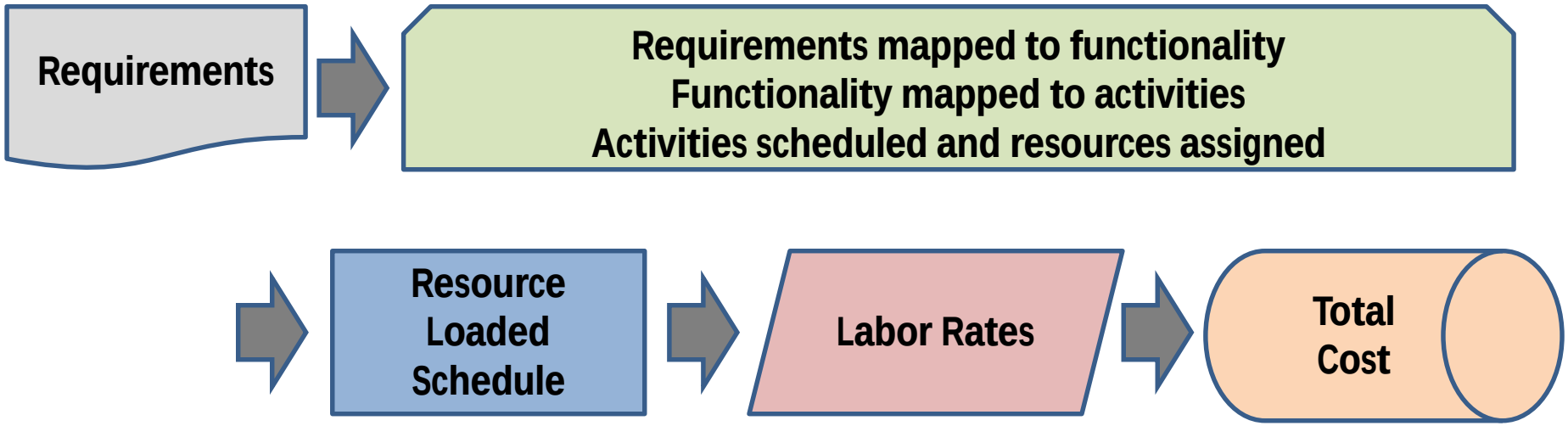
Functionality (cont'd)



Functionality (cont'd)

- Requirements mapped to functions which are mapped to the build plan or story boards
- Activities for each function created in project schedule
- Activities designed for five-to-ten FTEs
- FTEs estimated for each activity
- Labor rates applied to resource-loaded schedule
- JCL easily calculated from resource-loaded schedule
- Risks limited to labor rates, FTE counts, and length of activity schedule
- Risks are relatable to monitoring and controlling project management phase
- Activities defined in small enough increments to keep each activity under ten FTEs, making SME FTE estimates for reliable

Functionality (cont'd)



- **Sources of Uncertainty**
  - FTE counts
  - Labor rates
  - Schedule inputs

Common project management risks
Facilitates a Fully Integrated Cost Schedule Method (FICSM) and JCL
Aligns with project management principles

Summary

- All software estimates are founded in labor costs; the key is relating requirements to effort
- SLOC-based software estimating is subjective, difficult to cross-check, and has many risks not easily relatable to project management principles
- Messages are well understood and CER generation results in predictive intervals for risk assessments
- FP counting can be automated to reduce reliance on SME input
- Functionality-based estimating most directly relates to labor requirements, is necessarily integrated into schedule for JCL, and simplifies risk reduction and mitigation

Pre-MS A	Automated FPs
MS B	Functionality
Post-MS C	Messages

New processes improve uncertainty, reduce reliance on SME inputs, and incorporate FICSM and JCL

QUESTIONS?