Collaborative Requirements Scoring:
An Innovative Approach for Sizing Software Projects

Blaze Smallwood

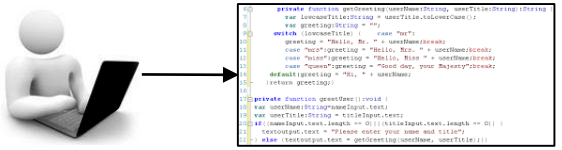
ICEAA Conference 2016

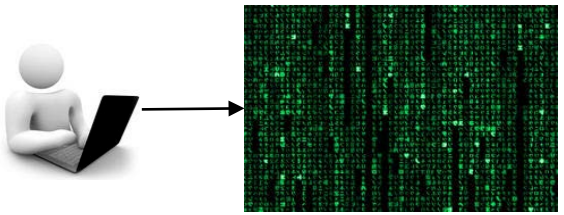
Table Of Contents

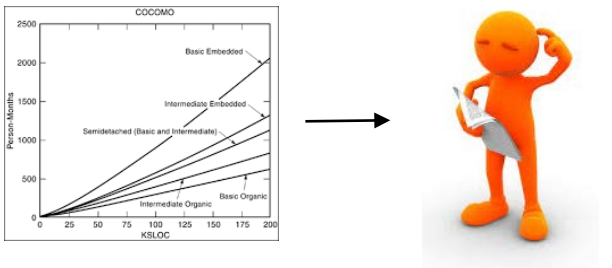
Table Of Contents

Existing software cost estimating techniques present various challenges, particularly when applying to modern systems with limited analogs and custom application requirements

- Estimating size using physical/functional measures is often non-intuitive to program engineers, particularly when estimating newer, lower-level requirements
- Accounting for non/cross-functional and/or derived requirements can be challenging
- Explaining to PMs how estimates are derived based on physical/functional sizing can be difficult
- Estimates are typically difficult to break out at lower levels of detail, hindering trade space analysis
- When estimates are wrong, hard to diagnose why







Traditional methods work well for many programs, but not all; some could use an alternative approach based on current software practices

Table Of Contents

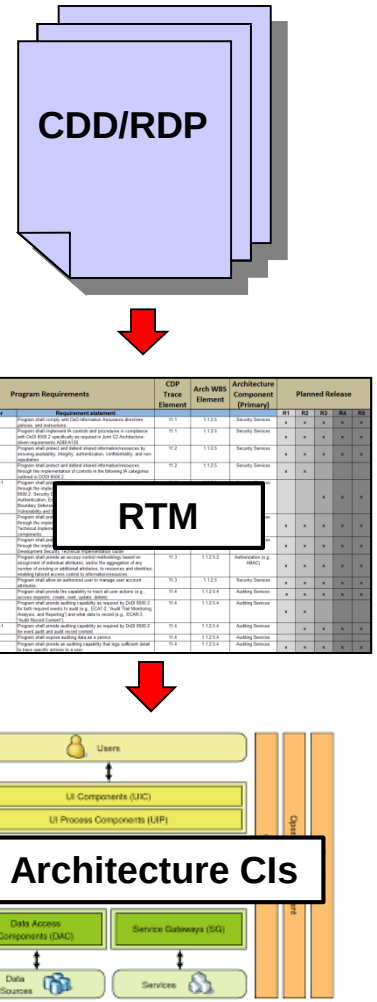
Table Of Contents

Step 1: Identify Appropriate Requirements to Size

Identifying appropriate requirements source and level of detail is essential to any sizing estimate

- Foundation would be functional/architectural requirements in program requirements document (e.g. Capability Definition Document (**CDD**) or Requirements Definition Package (**RDP**))
  - If **non-functional/cross-functional requirements** (architectural, usability, reliability, etc) are not captured in requirements document, they **can be identified and estimated during the sizing process**
- Requirements from CDDs or RDPs are often captured in a Requirements Traceability Matrix (**RTM**)
- Ideally, RTM would be used to map functional requirements to **components** in the intended **architecture**
  - Much more intuitive for SW engineers to estimate building components in an architecture

Ideal Requirements Flow



Best Practices for Requirements Identification

- Get agreement from estimate stakeholders (program manager, cost analyst, engineers) what requirements will be estimated and at what level
  - Need to identify the level of detail in requirements that fit time allocated to sizing and the appropriate units of measure (points, person-days, person-months)
- If documented requirements are all functional, work with engineers to identify non/cross-functional requirements/constraints before estimating session
- Identify trade space requirements up-front to inform later CAIV analysis
- Get Excel formats of requirements lists or matrices to help in building estimating template

Table Of Contents

Scoring sessions require upfront planning and coordination to be successful

- Identifying appropriate participants is key
  - "Scorers": Software engineering SMEs that will be providing estimates
    - A panel of four or five is ideal; minimum of three
    - Should be a mix of SMEs; at least a couple with experience in systems similar to the one being estimated, but independent SMEs add value, as well
  - Facilitator(s): Person who runs the meeting, records scores and assumptions, facilitates discussions, takes care of admin items
    - Ideally, two people – can be a lot for one person to handle
  - Other support SMEs: People knowledgeable about the program that can help scorers better understand requirements and constraints
    - Requirements analysts, test engineers, cyber experts, past users of similar system
- Coordinate schedules for participants and facilities early

Developing scoring materials and thinking through scoring process in advance will help the actual session go more smoothly

- Develop scoring template based on requirements to be estimated

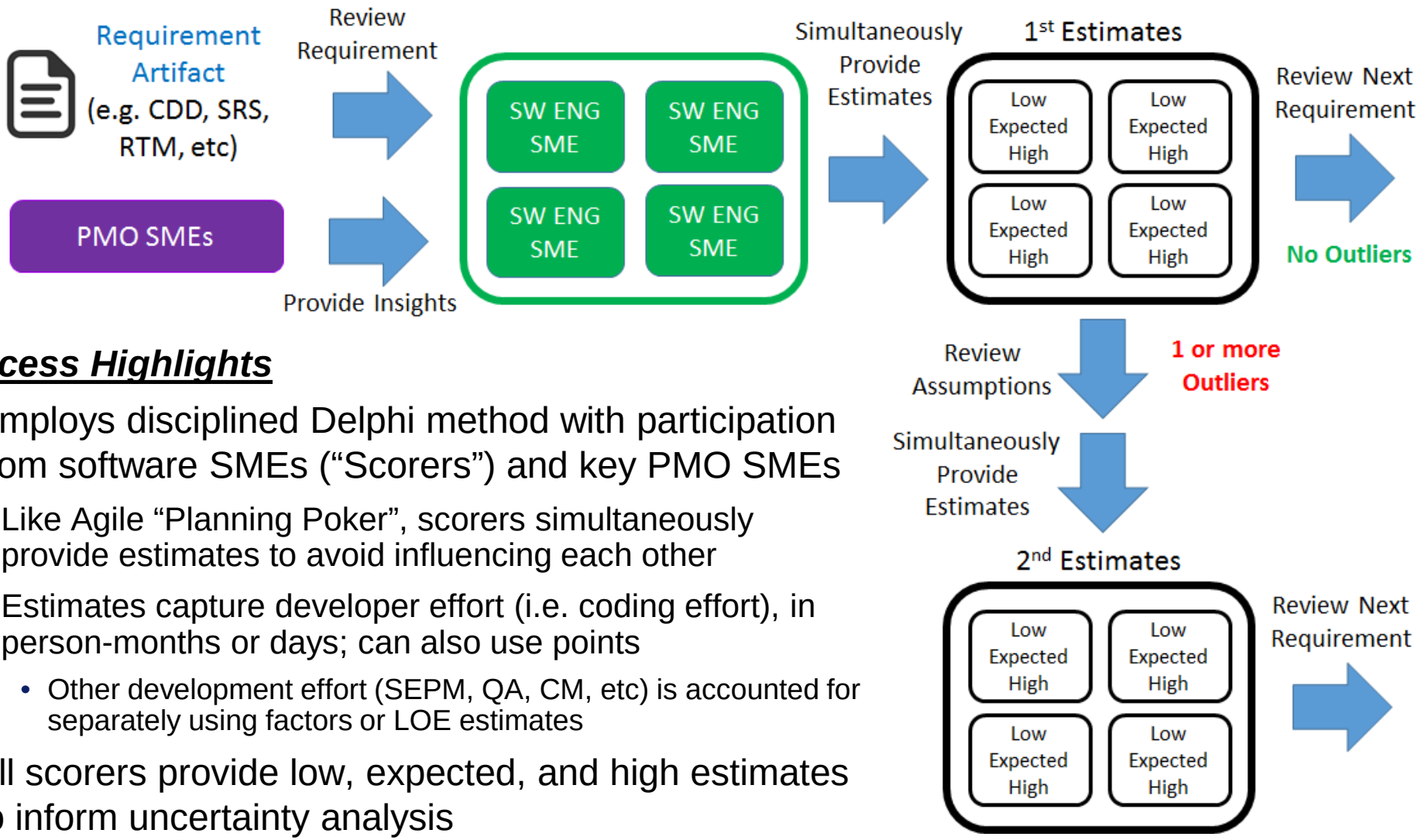
		Round 1									Round 2			
		Scorer 1			Scorer 2			Scorer 3			1	2	3	
Req #	Req. Description	Exp.	Low	High	Exp.	Low	High	Exp.	Low	High				Comments
1.1.1	The system shall enable the user to . . .													
1.1.2	The system shall enable the user to . . .													
1.2.1	The system shall enable the user to . . .													
1.2.2	The system shall enable the user to . . .													
1.2.3	The system shall enable the user to . . .													
2.1.1	The system shall enable the user to . . .													
2.1.2	The system shall enable the user to . . .													
2.1.3	The system shall enable the user to . . .													
X.X.X	The system shall enable the user to . . .													

- Coordinate systems or materials needed to conduct scoring; session requires scorers to provide estimates anonymously and simultaneously
  - Decision analysis system, like Expert Choice
  - Agile Planning Poker app
  - Real-time collaboration tools, like Google Docs or Real Time Board
  - Screen sharing, like WebEx or Defense Collaboration Services (DCS)
  - White boards

Table Of Contents

Scoring process helps objectify subjective inputs and documents assumptions for each estimate

Step 3: Execute Scoring Session



Process Highlights

- Employs disciplined Delphi method with participation from software SMEs ("Scorers") and key PMO SMEs
  - Like Agile "Planning Poker", scorers simultaneously provide estimates to avoid influencing each other
  - Estimates capture developer effort (i.e. coding effort), in person-months or days; can also use points
    - Other development effort (SEPM, QA, CM, etc) is accounted for separately using factors or LOE estimates
- All scorers provide low, expected, and high estimates to inform uncertainty analysis

Best Practices for Executing Scoring Session

- Baseline all participants upfront on ground rules and assumptions
  - Ensure everyone is operating off of the same overarching technical assumptions
  - Agree on the scope of estimates to be provided; typically, developer effort from design to code unit test
  - Discussion before scores should be limited to technical assumptions only
    - No value statements -> “This is easy, should be minimal effort”
  - Second round is needed when one or more outlier scores
- Ensure everyone is clear on time constraints to stay on schedule
  - Calculate benchmark requirements to measure progress against time plan
- Capture all assumptions for each requirement in scoring spreadsheet; capture any other thoughts on white boards or smartboards & take pictures
- Use white board to capture “parking lot” items; could inform additional items to score, like cross-functional or derived requirements

Step 3: Execute Scoring Session

Outputs of a successful scoring session

- Fully populated scoring template with all scores and documentation comments

		Round 1									Round 2			
		Scorer 1			Scorer 2			Scorer 3			1	2	3	
Req #	Req. Description	Exp.	Low	High	Exp.	Low	High	Exp.	Low	High				Comments
1.1.1	The system shall enable the user to . . .	8	6	10	8	6	10	6	4	8				Need to build API and widget XYZ
1.1.2	The system shall enable the user to . . .	6	4	8	8	6	10	6	4	8				Need to edit data handling APIs
1.2.1	The system shall enable the user to . . .	4	3	5	1	1	1	5	4	7				Need to build presentation widget
1.2.2	The system shall enable the user to . . .	7	5	9	6	4	8	10	7	13				Need to integtrate XYZ COTS tool
1.2.3	The system shall enable the user to . . .	7	5	9	10	7	13	4	3	5				Need to implement XYZ interface
2.1.1	The system shall enable the user to . . .	2	1	3	4	3	5	1	1	1				Need to implement XYZ function in data layer
2.1.2	The system shall enable the user to . . .													Effort included in estimate above
2.1.3	The system shall enable the user to . . .	3	2	4	1	1	1	3	2	4				Need to build presentation widget
[illegible]	[illegible]	[illegible]												
[illegible]	[illegible]	[illegible]												
[illegible]	[illegible]	[illegible]												
[illegible]	[illegible]	[illegible]												
[illegible]	[illegible]	[illegible]												
X.X.X	The system shall enable the user to . . .	2	1	3	7	5	9	9	6	12				Need to build API and widget XYZ

- Additional information captured in the room during the session, such as sketches or assumptions on a white board or smart board captures

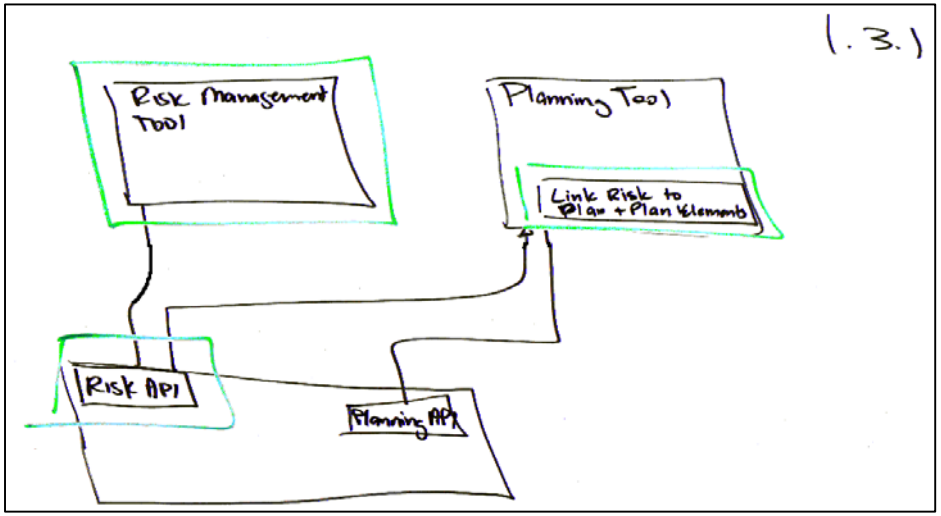
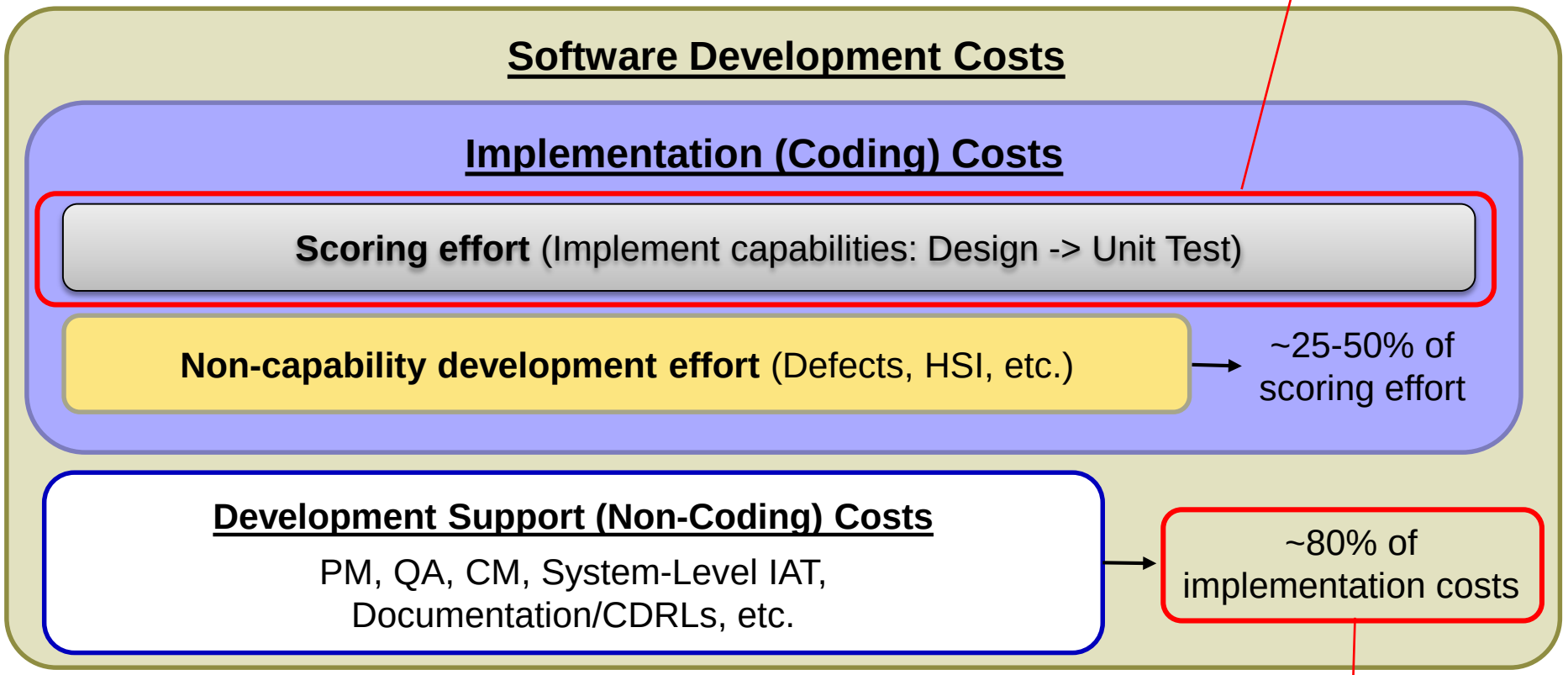


Table Of Contents

Other costs must be added to the scoring outputs to derive full software development cost estimate

Estimated based on scoring session outputs; provides foundation for the rest of the SW development estimate

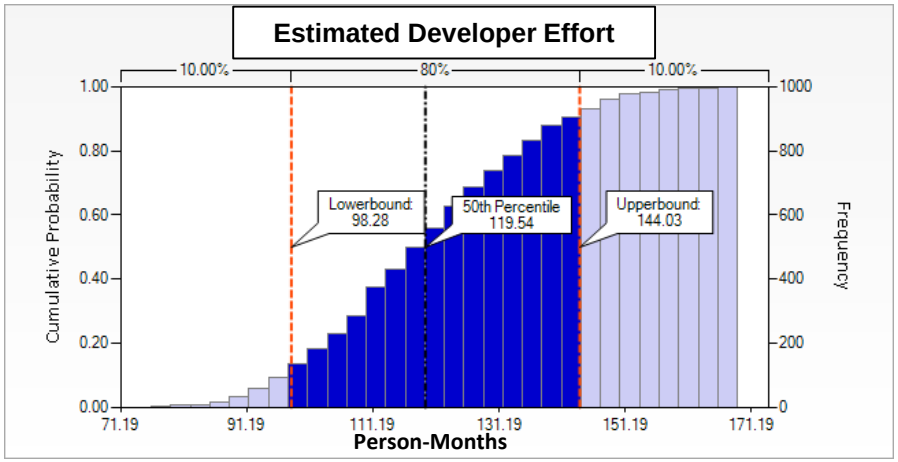


Can be estimated using a factor like this or using a LOE build-up

Ranges provided during scoring analysis provide solid inputs for robust uncertainty analysis

- Ranges provided by multiple scorers provide many possibilities for uncertainty bounds
  - Average of Expected
  - Min of Low, Max of High
  - Average of High as Expected
- Uncertainty can be applied at whatever level in requirements desired

		Round 1									Round 2		
		Scorer 1			Scorer 2			Scorer 3			1	2	3
Req #	Req. Description	Exp.	Low	High	Exp.	Low	High	Exp.	Low	High			
1.1.1	The system shall enable the user to . . .	8	6	10	8	6	10	6	4	8			
1.1.2	The system shall enable the user to . . .	6	4	8	8	6	10	6	4	8			
1.2.1	The system shall enable the user to . . .	4	3	5	1	1	1	5	4	7			
1.2.2	The system shall enable the user to . . .	7	5	9	6	4	8	10	7	13			
1.2.3	The system shall enable the user to . . .	7	5	9	10	7	13	4	3	5			
2.1.1	The system shall enable the user to . . .	2	1	3	4	3	5	1	1	1			
2.1.2	The system shall enable the user to . . .												
2.1.3	The system shall enable the user to . . .	3	2	4	1	1	1	3	2	4			
X.X.X	The system shall enable the user to . . .	2	1	3	7	5	9	9	6	12			



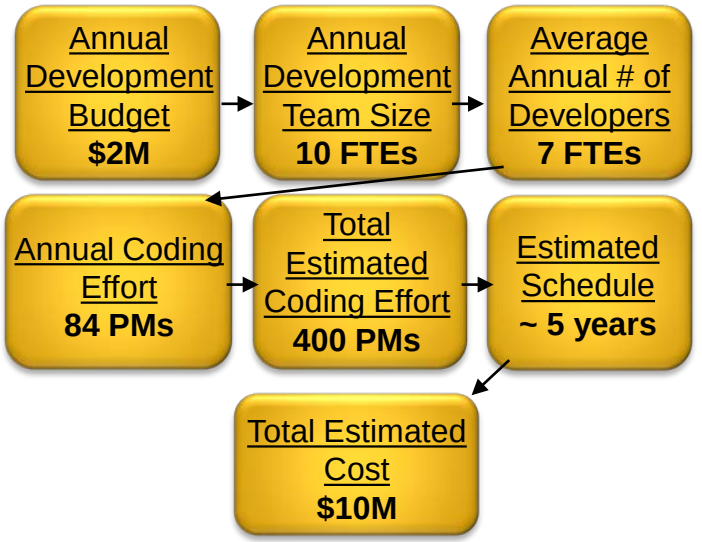
Uncertainty analysis is a pivotal step to bound cost and schedule estimates

Uncertainty adjusted developer estimates can be used to build up total estimate using multiple methodologies

- Variable schedule/CAIV driven methodology based on annual budget constraints
  - Annual budget determines team size and number of developers
  - # of developers coupled with scoring session effort estimates determine total estimated schedule
  - Non-coding effort can be added along that schedule using factors or LOE
- Fixed schedule driven methodology based on schedule constraints
  - Targeted schedule determines how developer effort gets spread and required # of developers
  - Non-coding effort can be added using factors or LOE
  - Requires sanity check on required development team size for reasonableness

Examples

Variable Schedule / Fixed Annual Cost



Fixed Schedule / Variable Annual Cost

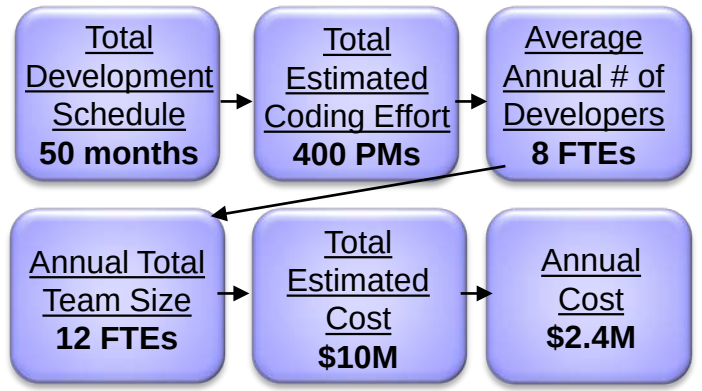


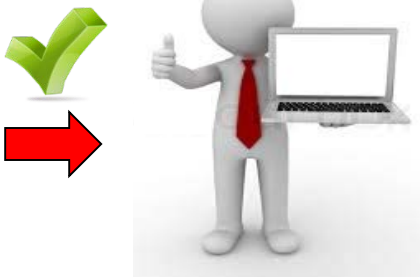
Table Of Contents

Utilizing this process to estimate size of a software project has various benefits and a few challenges

Benefits
More intuitive scope sizing methodology
Sizing explicitly relates complexity to effort
Systematic scoring process
Scoring ranges inform robust uncertainty
Allows for trade-off analysis at requirement level

Challenges
Subjective sizing inputs; limited analogous data
New type of cost model required
Sizable effort/coordination to run scoring session

		Round 1								
		Scorer 1			Scorer 2			Scorer 3		
Req #	Req. Description	Exp.	Low	High	Exp.	Low	High	Exp.	Low	High
1.1.1	The system shall enable the user to …	8	6	10	8	6	10	6	4	8
1.1.2	The system shall enable the user to	6	4	8	8	6	10	6	4	8
1.2.1	The system shall enable the user to …	4	3	5	1	1	1	5	4	7
1.2.2	The system shall enable the user to …	7	5	9	6	4	8	10	7	13
1.2.3	The system shall enable the user to …	7	5	9	10	7	13	4	3	5
2.1.1	The system shall enable the user to …	2	1	3	4	3	5	1	1	1
2.1.2	The system shall enable the user to									
2.1.3	The system shall enable the user to …	3	2	4	1	1	1	3	2	4
XXX	The system shall enable the user to	2	1	3	7	5	9	9	6	12



$$PM = A \cdot Size^{E} \cdot \prod_{i=1}^{n} EM_i$$

For many projects, the benefits outweigh the challenges, and challenges can be mitigated

Table Of Contents

Collaborative requirements estimating is a viable new methodology for informing cost estimates for software projects

- While traditional methodologies are still viable for various types of software projects, this new methodology is viable for newer projects with limited analogs and new requirements
  - Scoring methodology is intuitive to software engineers and aligns with how software teams estimate, plan, and execute work
  - Disciplined scoring process attempts to add objectivity and documentation to subjective inputs
  - Scoring ranges enable detailed uncertainty analysis
  - Estimates at low requirement levels enable detailed scope trade-off analysis
  - Easy to explain to decision makers and diagnose estimating error

Questions?

For further information . . .

Blaze Smallwood
Associate

Booz | Allen | Hamilton

Booz Allen Hamilton Inc.
Office 309.359.3160
Mobile 619.850.6123
smallwood_blaze@bah.com