Agile MVPs and EVM

Implications of truly agile Minimally Viable Products (MVPs) on the use of EVM

SOUTHERN ILLINOIS UNIVERSITY
EDWARDSVILLE
SCHOOL OF BUSINESS

About…

Dale Shelton

- US Transportation Command, Program Executive Office
- Chief, Systems and Services Division
- PM, Global Household Goods Contract

Leads an acquisition division that includes defense business systems (move.mil, MilMove prototype, and the Defense Personal Property System) and Services Category (SCAT) 1 and 2 Service Acquisitions for worldwide household goods shipping and storage, and vehicle shipping. This division supports the Defense Personal Property Program's multibillion dollar relocation enterprise.

Agenda

- Earned Value Management (EVM) Summary
- EVM and Agile
- Emerging DoD skepticism on wrapping EVM around Agile
- Discussion about options to get decent estimates and trade space without full analysis

Earned Value Management (EVM)

- Understand current project state relative to planned state
- Predict completion cost and date
- Assumptions
  - Full scope is understood
  - Value equals cost
  - Work package estimates are accurate
  - Work package sequence does not change

EVM in the Government

- Required for Cost Contracts > $20M Lifecycle, scrutinized over $50M
- Drives suboptimal contract types?
  - Commercial (therefore FFP) to avoid EVM
  - Waterfall so EVM works as intended
- How do I avoid it?
  - Approved Determination and Finding
  - Defense Personal Property System Example

EVM and Agile

- PMI Agile Practice Guide, "Traditional EVM metrics like schedule performance index (SPI) and cost performance index (CPI) can be easily translated into agile terms."
  - SPI = ratio of completed to planned story points
  - CPI = ratio of earned value (completed features) to the actual costs

Is Translation Easy?

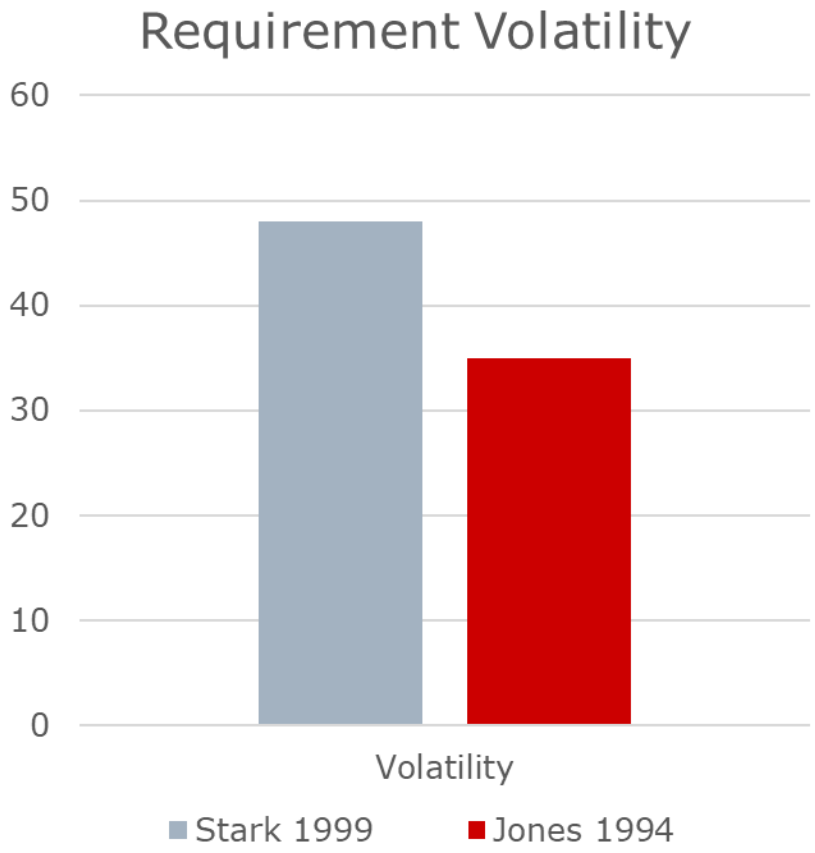
- If it is, you may not "agiling" right
  - How can you know the size without 100% plan?
  - Distance skews perspective
  - Focus on the MPV
  - Frequent design and priority changes
- Ratio of completed features to actual costs
  - Capacity costs tend to be flat, based on team size
  - Completed work loses meaning without knowing how much work is left
  - But this isn't completely off, it's basically burn down

Conflict between EVM and Agile Values

- ***Customer collaboration*** *over contract negotiation*
- ***Responding to change*** *over following a plan*

- EVM relies on Work Packages' Cost and Sequence
  - Re-sequencing skews predictions
  - Dropped, added, or redefined packages
  - Rebaselining
- Can we resolve tension between EVM's hunger for WBS and agile's embrace of change?

Requirement Volatility is a Historic Problem



- Stark Study of 44 releases found 48% to 600% (1999)
- Jones Study of 60 projects found 35% (1994)
- Among them, 33% to 50% delivered requirements that were not part of the original plan
- Jones found requirements change at 1% per month

Agile Embraces Requirement Volatility

- ***Customer collaboration*** *over contract negotiation*
- ***Responding to change*** *over following a plan*

- Agile Cost, Quality, and Schedule are fixed
  - Capability varies to fit those constraints
- Frequent user interaction refines what's possible
  - Further increases requirement volatility
- How deep / wide should elicitation be?
  - T-Shirt Sizing of road-mapped features vice pointing
  - Just-in-time story design, don't design it twice
  - Estimates based on complexity as defined by the individuals conducting the analysis
  - Teams' velocity and pointing evolve over time

Concern Gathering on EVM vs Agile

- SEI, Agile Metrics: Progress Monitoring of Agile Contractors, 2014
  - [EVM] is foreign to a typical Agile implementation
- NDAA 2018
  - Sec Def select at least four agile projects and omit EVM, IMS, IMP, Tech Docs, etc.
- Software is Never Done (Defense Innovation Board, 2019)
  - Revise DFARS Subpart 234.201, DoDI 5000.02 Table 8, and OMB Circular A-11 to remove EVM requirement.

Right tool for the job

- Don’t design too early
  - Requirements will change anyway
- Which agile metrics are best predictors?
  - Burn up/down, roadmap, etc.
- But if we can’t estimate, how do we budget and with what level of confidence?

Agile Estimating Discussion

- Engineer only enough to get to the next step
  - How sure can we be about what it takes to get "done"?
- Cocomo II – can we get enough function points?
- Use Case Estimating – can we get enough fidelity?
- MoSCoW Estimating – can this establish objective and thresholds levels?