The Budding SV3 – Estimating the Cost of Architectural Growth Early in the Life Cycle

*LTC Matthew Dabkowski**
Dr. Ricardo Valerdi

Department of Systems & Industrial Engineering
College of Engineering
University of Arizona

* mfd1@email.arizona.edu

Disclaimer: This material is based upon work supported by the Naval Postgraduate School Acquisition Research Program under Grant No. N00244-13-1-0032, and the Office of the Secretary of Defense. The views expressed in written materials or publications, and/or made by speakers, moderators, and presenters, do not necessarily reflect the official policies of the Naval Postgraduate School nor does mention of trade names, commercial practices, or organizations imply endorsement by the U.S. Government.

Agenda

- Purpose, Question, and Contribution
- Political and Acquisition Background
- Peril of Pre-MS A Cost Estimation
- Systems Engineering (SE) – A Discipline with Promise
- Constructive Systems Engineering Cost Model – A Tool for Costing Architectural Complexity
- Network Science – A Mechanism for Generating Unforeseen Architectural Growth
- Simulating Growth and Estimating Cost
- Limitations
- Future work
- Questions

Overarching Purpose: To transform Model-Based System Engineering (MBSE) artifacts into computational knowledge that can be leveraged early in the system lifecycle when uncertainty is high and confidence is low

↓

Focused Question: Can parametric cost estimation, in conjunction with DoD Architecture Framework (DoDAF) models, capture the monetary impact of architectural changes early in the system lifecycle?

↓

Principal Contribution: A network science-based algorithm for estimating the cost of unforeseen architectural growth

Flirting with a cliff

Sequence of Events

Situation

Federal spending unsustainable / national debt exploding

Solution

August 2, 2011: POTUS signs Budget Control Act into law with *threat* of sequestration

Action (or Inaction)

Congress fails to pass a deficit reduction bill

Reaction

March 1, 2013: As promised, POTUS implements cuts

Impact on DoD Funding

- **Unanticipated 23% reduction in Fiscal Year 2013 budget**
- **Combined $492 billion loss in funding over next 10 years**

Second Order Effects

- **Furloughs**
- **Reduced production**
- **Difficult modernization decisions . . .**

Single Largest Impact

“single largest impact of sequestration and current budgetary unknowns is [their effect on] . . . the **meticulous cost and schedule planning** mandated in numerous public laws and DoD acquisition policy directives”*

Dr. William LaPlante
Principal Deputy Assistant Secretary
of the Air Force (Acquisition)

LTG Michael Moeller
USAF Deputy Chief of Staff,
Strategic Plans and Programs

* *On impacts of a continuing resolution and sequestration on acquisition, programming, and the industrial base: Hearing before the Subcommittee on Tactical Air and Land Forces, Committee on Armed Services*, House of Representatives, 113th Cong., 1st Sess., 12 (2013).

We find ourselves in challenging times

- Even before the cuts, **meticulous** cost planning was tough
  - Between 1997 and 2009, 47 major defense acquisition programs (MDAPs) experienced cost overruns of at least 15% or 30% over their current or original baseline estimates*
- Conditions have not improved

The Requirement

In an uncertain environment, the need to plan well is paramount

The Reality

Budget uncertain and presumably shrinking

Uncertain funding frustrates cost planning

Cost planning already difficult

* Government Accountability Office. (2011). *DOD COST OVERRUNS: Trends in Nunn-McCurdy Breaches and Tools to Manage Weapon Systems Acquisition Costs.*

Leaning forward in the foxhole

- Despite recent challenges, Congress recognized need for change in 2009 with the Weapon Systems Acquisition Reform Act (WSARA)
- 4 organizational changes and 7 procedural adjustments, including:
  - Increased rigor of Pre-Milestone A (Pre-MS A) cost analysis
  - Greater emphasis on integration and design risk by (1) establishing Performance Assessments and Root Cause Analysis (PARCA) official and (2) requiring the Director of Defense Research and Engineering (DDR&E) to develop standards to measure integration risk
- Majority of program's future, life-cycle costs are often (perhaps unwittingly) committed in preliminary design phase

Today's design decisions, driven by current requirements, determine tomorrow's debts ⇒ make better design decisions now

Pre-MS A cost estimation is fraught with peril

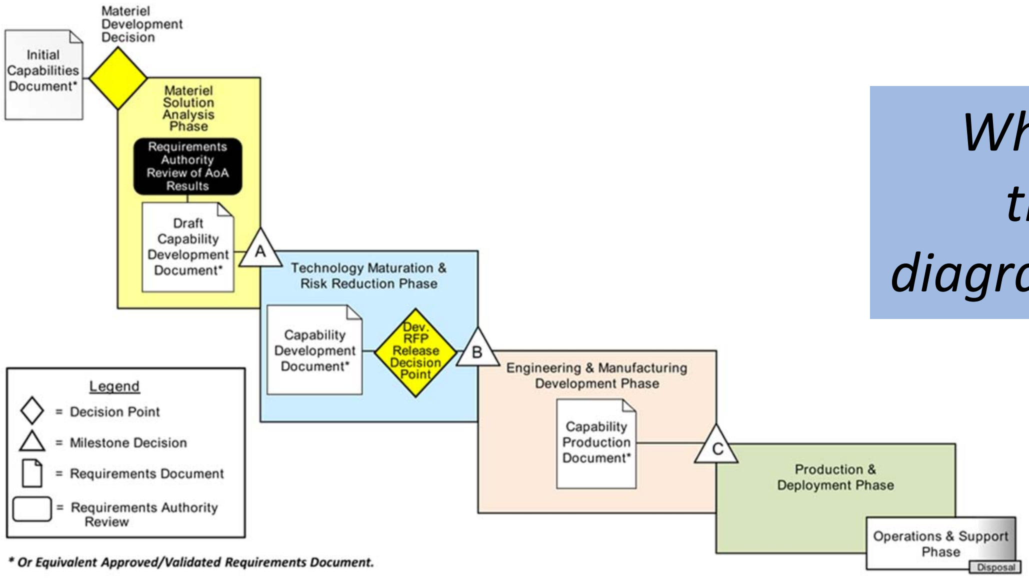
- Early life cycle characterized by uncertainty*
  - Requirements change: Average of 28% of a system's baseline requirements will change
  - Changes often come late: Roughly 43% of changes occur in the development phase
  - More likely to add than take away: 86% of changes manifest as modification of an existing requirement or addition of a new requirement
- Post MS-B requirement changes account for an average of 21.5% of a program's total cost growth over its MS B estimate†

* Peña, M., & Valerdi, R. (2014, in press). Characterizing the Impact of Requirements Volatility on Systems Engineering Effort. *Systems Engineering*.

† Bolten, J. G., Leonard, R. S., Arena M. V., Younossi, O., & Sollinger, J. M. (2008). *Sources of Weapon System Cost Growth - Analysis of 35 Major Defense Acquisition Programs*. Santa Monica, CA: RAND Corporation.

But More Information is Available Earlier

- Interim DoDI 5000.02 now requires a “draft” or DoD component-approved Capability Development Document (CDD) prior to MS A
- CDDs must contain 25 of the 31 DoD Architecture Framework (DoDAF) models ever required by JCIDS . . . the same set required by the Capability Production Document (CPD)



* Or Equivalent Approved/Validated Requirements Document.

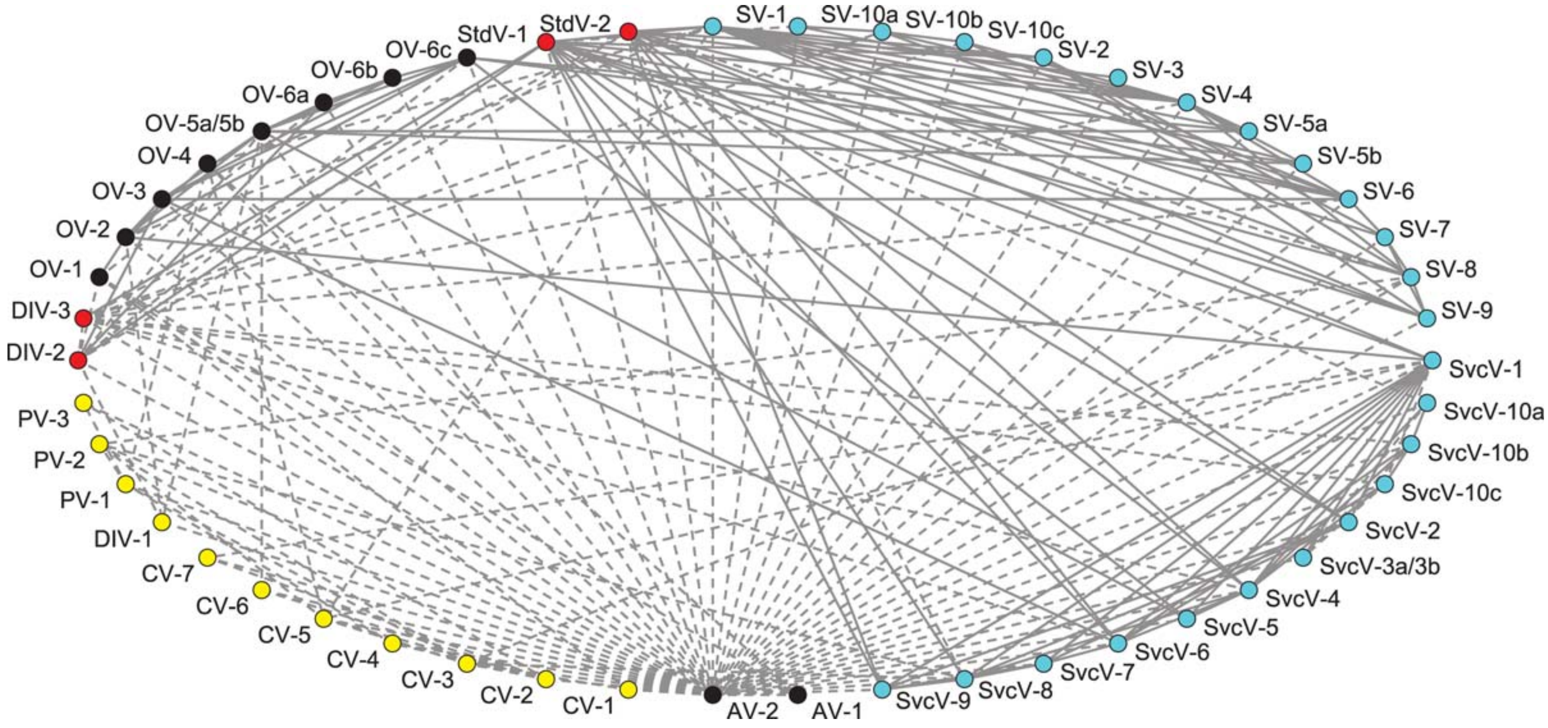
What, if anything, can these draft DoDAF diagrams tell us about cost?

What is DoDAF Besides Mandatory?

- DoDAF is . . .
  - A conceptual modeling paradigm that provides common understanding*
  - In its third major revision (v1.0 → v1.5 → v2.0 (v2.02))
  - Represented as 52 models organized into 8 collections (viewpoints)
  - Designed to be data versus product-driven
  - Meant to be “fit-for-purpose” versus rigid
  - Serving as the DoD’s foundation for Model-Based System Engineering
  - Implemented in sophisticated SE tools (Atego’s Artisan Studio, IBM’s Rational Systems Architect, Vitech’s CORE, etc.)
  - Potentially overwhelming

* http://dodcio.defense.gov/dodaf20/dodaf20_background.aspx

DoDAF v2.02 as a Network



Legend

- ● - models from v1.5 replicated in v2.02
- 🔴 - models from v1.5 renamed in v2.02
- 🔵 - models from v1.5 bifurcated in v2.02
- 🟡 - new models in v2.02
- — - explicit connections from v1.5 manual
- - - - - implicit connections from v2.02 manual

AV - all (2 models)
CV - capability (7 models)
DIV - data and information (3 models)
OV - operational (9 models)

PV - project (3 models)
StdV - standards (2 models)
SvcV - services (13 models)
SV - systems (13 models)

Why do we need DoDAF Pre-MS A?

- DoD procurements are often large, complex, and expensive

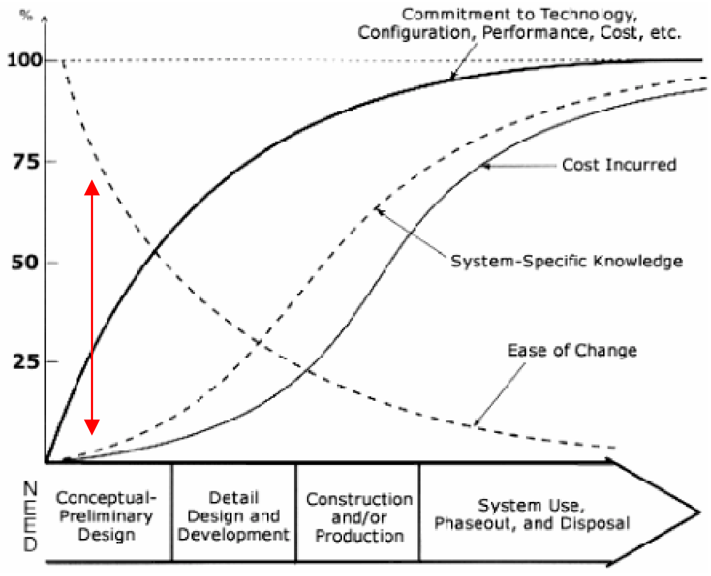
F-35 Joint Strike Fighter (JSF)

- 3 major contractors (LM, NG, BAE)[*]
- 9 partner nations[*]
- 24 million lines of code[Δ]
- 10^5 interfaces[†]
- Total lifecycle cost ≈ $1.51T[‡]

- When changing complex systems, earlier is easier, but change requires knowledge[§]

DoDAF Pre-MS A helps close this gap

Figure 2.11 from Blanchard & Fabrycky (1998).



---

[*] http://www.jsf.mil/f35/

[Δ] Hagen, C. & Sorenson, J. (2013). Delivering Military Software Affordably. *Defense AT&L Magazine, March-April 2013*, 30-34.

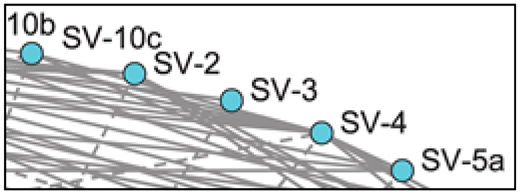
[†] Becz et al. (2010). Design System for Managing Complexity in Aerospace Systems. Paper presented at the 2010 AIAA ATIO/ISSMO Conference, Fort Worth, Texas.

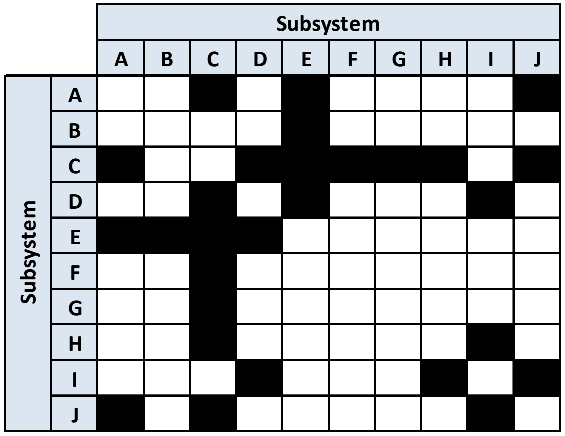
[‡] http://articles.chicagotribune.com/2012-04-02/news/sns-rt-us-lockheed-fighterbre8310wb-20120402_1_problems-or-cost-increases-technical-problems-or-cost-f-35.

[§] Blanchard, B., & Fabrycky W. (1998). *Systems engineering and analysis* (3rd ed.). Upper Saddle River, NJ: Prentice Hall.

Systems Engineering (SE) – A Discipline with Promise

- Consider the SV-3: Systems-Systems Matrix, which compactly captures how the subsystems of a larger system connect
- SV3 falls within purview of SE
- SE holds promise for controlling cost
  - *DoD Cost Overruns*: “early and continued SE analysis” helps contain cost growth
  - *2008 National Research Council (NRC) report*: “application of SE to decisions made in the pre-Milestone A period is critical to avoiding (or at least minimizing) cost and schedule overruns”
  - *ROI of SE:* ROI for greater SE effort as high as 8:1
  - *WSARA 2009*: Added a Director of SE to the USD(AT&L)’s staff





		Subsystem									
		A	B	C	D	E	F	G	H	I	J
Subsystem	A			■		■					■
	B					■					
	C	■			■	■	■	■	■		■
	D			■		■				■	
	E	■	■	■	■						
	F			■							
	G			■							
	H			■						■	
	I				■				■		■
	J	■		■						■	

SV3

Constructive Systems Engineering Cost Model (COSYSMO) – A Tool for Costing Architectural Complexity

- Open academic parametric cost model
- CER incorporates *size of the system* and *SE effort* required

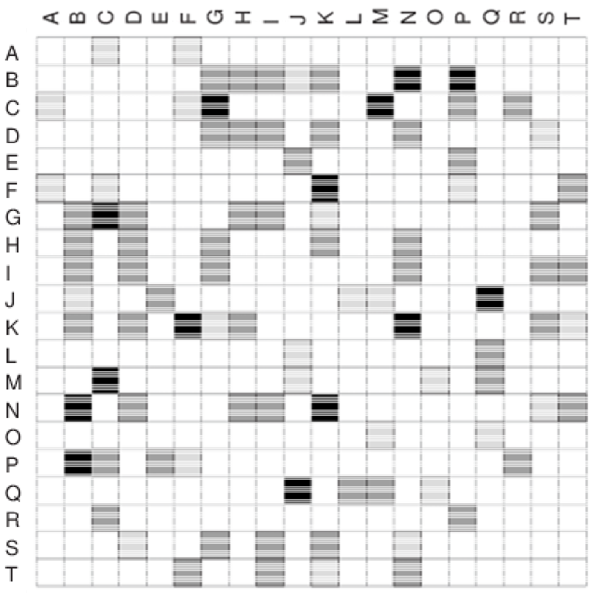
$$PM_{NS} = A \cdot \underbrace{\left(\sum_{i \in \{e,n,d\}} \sum_{k=1}^{4} w_{i_k} \Phi_{i_k} \right)^{E}}_{\text{size}} \cdot \underbrace{\prod_{j=1}^{14} EM_j}_{\text{effort}}$$

- SV-3 directly related to the "number of interfaces" size driver
- For complexity, all interfaces <u>are not</u> created equal

Interfaces	Complexity		
Characteristics	Easy	Nominal	Difficult
Message complexity	Simple	Moderate	Complex
Coupling level	Uncoupled	Loose	Tight
Stakeholder consensus	Strong	Moderate	Low
Behavior	Well behaved	Predictable	Emergent
Relative weight (w_{ik})	1.1	2.8	6.3

Hypothetical SV-3

- 20 subsystems with 47 interfaces of varying complexity
- Without loss of generality, assume there are . . .
  - 200 easy, 200 nominal, and 50 difficult requirements
  - 5 difficult critical algorithms
- Using additional w_{ik} and EM_j data,* apply CER to obtain an initial estimate of PM_{NS}



SV-3

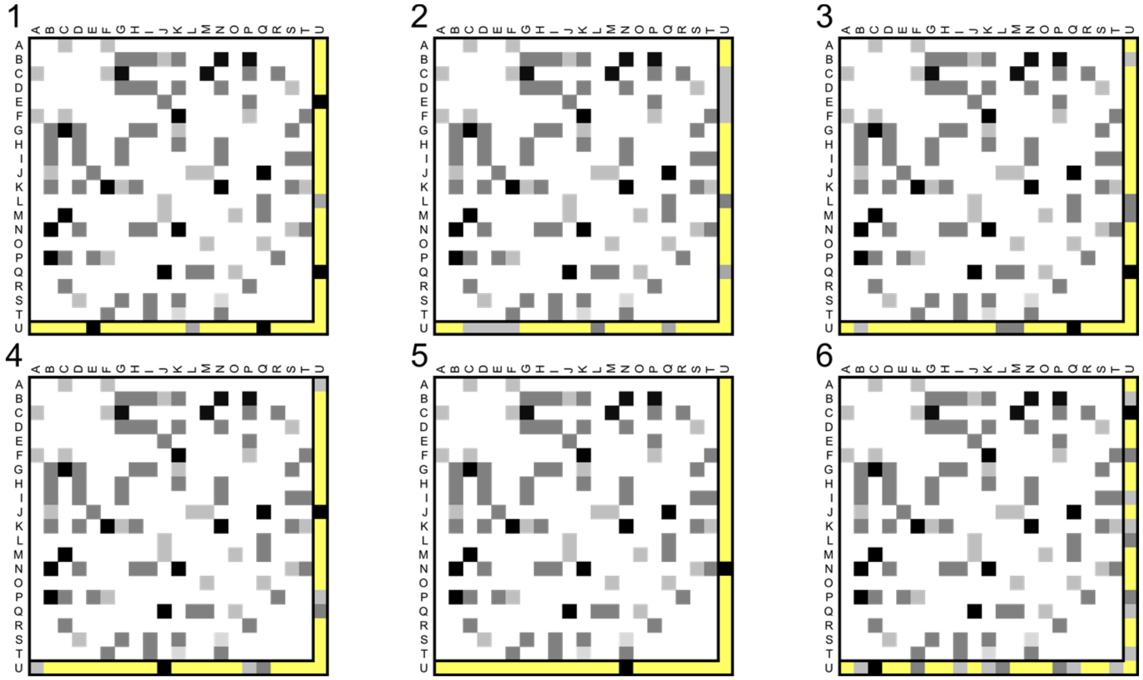
Interface Complexity

□ = Easy, ▨= Nominal, ■ = Difficult

$$PM_{NS} = 0.25 \cdot \left(\underbrace{(0.5 \times 200 + 1.0 \times 200 + 5.0 \times 50)}_{\text{requirements}} + \underbrace{(11.5 \times 5)}_{\text{algorithms}} + \underbrace{(1.1 \times 13 + 2.8 \times 27 + 6.3 \times 7)}_{\text{interfaces}} \right)^{1.06} \cdot 0.89 = 245.27$$

What about inevitable, unforeseen change?

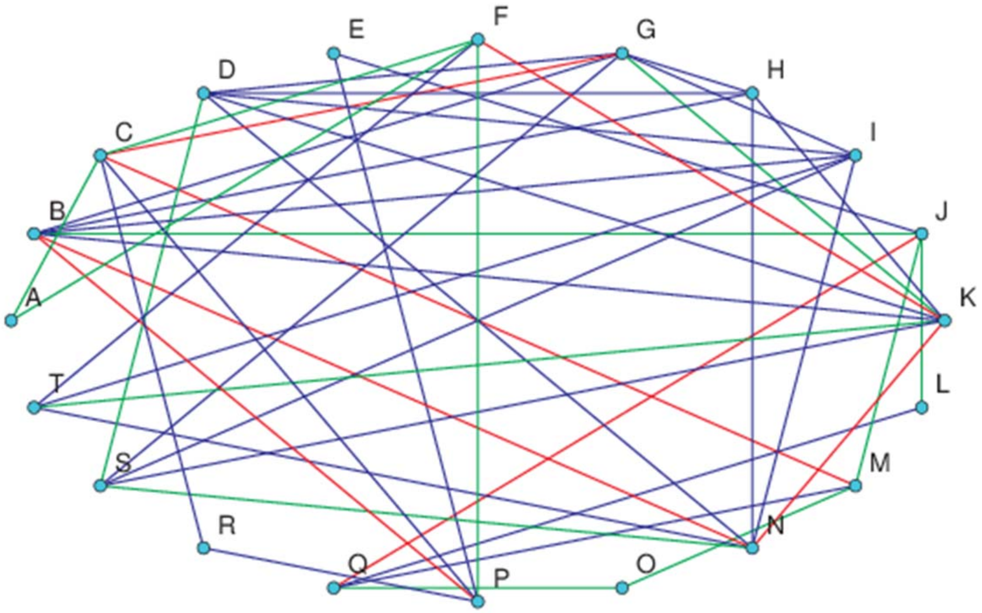
- 245.27 PM of SE effort are required, and, if each PM costs $20,000, this equates to $4,905,400
- But, this is Pre-MS A; <u>what happens when requirements change</u>?



What if we add a new subsystem to the architecture <u>without knowing</u> its purpose?

azengineering

The Analytical Task



Graphical Representation
of the SV-3

Interface Complexity

🟩 = Easy, 🟦 = Nominal, 🟥 = Difficult

*If we add a new subsystem **U**, how will it connect to the existing architecture?*

a) How many subsystems should **U** connect to (**degree**)?,

b) Given **U** connects to *d* subsystems, which *d* subsystems should it connect to (**adjacency**)?, and

c) Given **U** connects to a specific set of *d* subsystems, what should the complexity of these interfaces be (**weights**)?

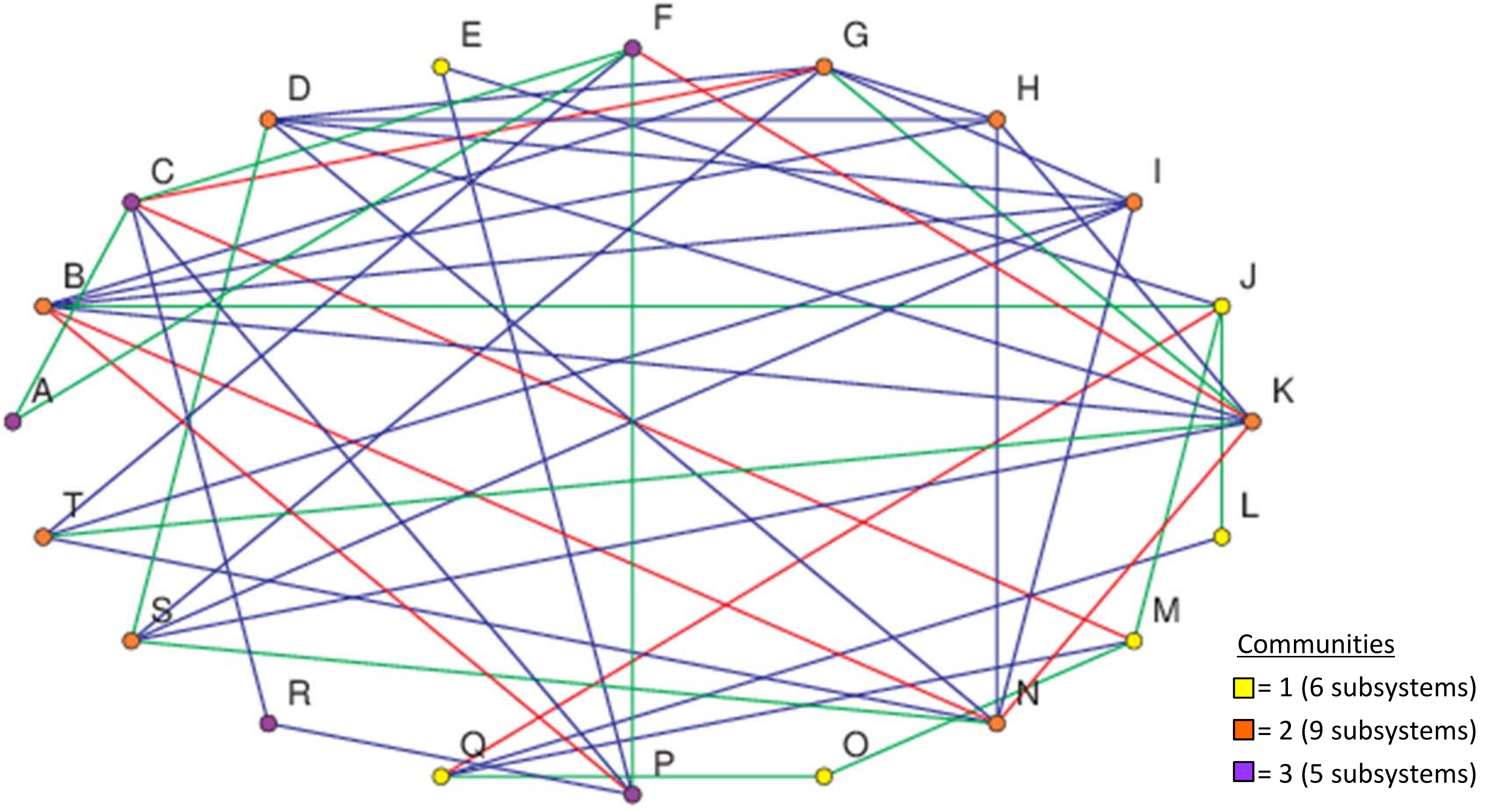
What will it cost?

Network Science – A Mechanism for Generating Unforeseen Architectural Growth

- Two fundamental assumptions
  1. Manner in which SEs architect complex systems matters
     - Partition: Search for "architectural communities" using Girvan-Newman
  2. Existing architecture foretells future architecture
     - Degree: Treat degree of **U** ($D_\mathbf{U}$) as a random variable with a probability mass function (pmf) equal to observed degree distribution of existing system ("rich-by-birth" effect)
     - Adjacency: Utilize Barabási–Albert preferential attachment (PA) model, where highly connected subsystems are more likely to interface with **U** ("rich-get-richer" effect); and
     - Weights: Model complexity of interface between **U** and subsystem ***i*** ($w_{i\mathbf{U}}$) as a conditional random variable, where pmf for $w_{i\mathbf{U}}$ equates to observed interface complexity distribution of subsystem ***i***

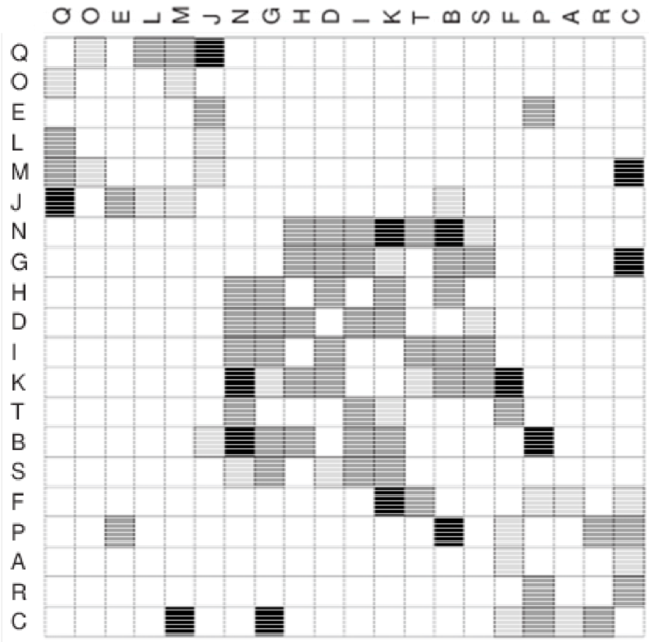
Identifying Architectural Communities

- Back to our hypothetical SV3 . . .
- 3 communities detected using Girvan-Newman



Isomorphic SV3

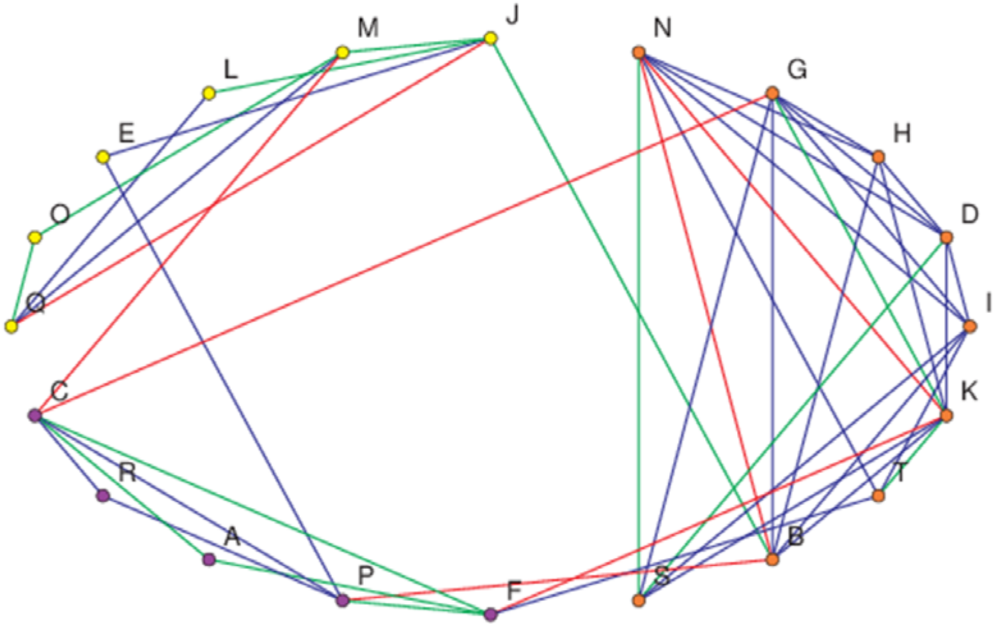
- Permuted SV3 reinforces the partitioning



Adjacency Matrix Representation
SV3 in DoDAF

<u>Interface Complexity</u>

□ = Easy, ▨= Nominal, ■ = Difficult



Graphical Representation

<u>Interface Complexity</u>

🟩 = Easy, 🟦= Nominal, 🟥 = Difficult

Simulating Growth and Estimating Cost

- Pseudo-code to estimate cost impact of adding **U**

```
For a specified number of iterations . . .
    1. Initialize the system as the current system
    2. Use Girvan-Newman to identify architectural communities
    3. Randomly assign U to community $j$
```

Intracommunity growth

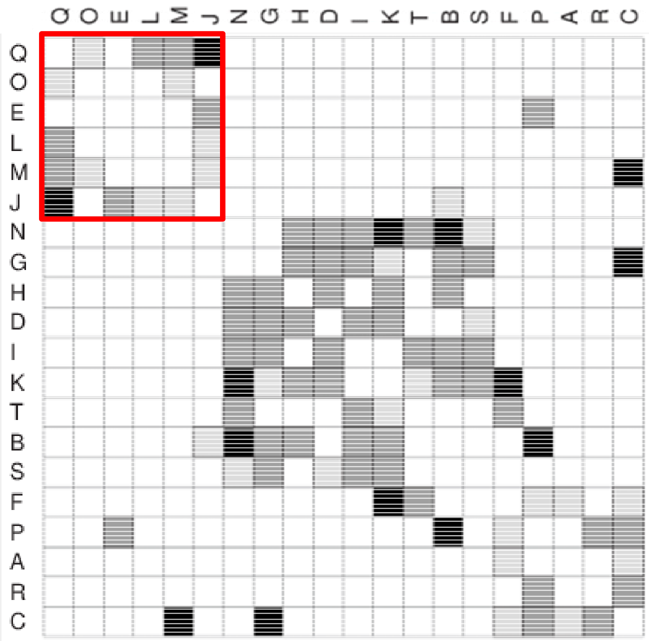
```
    4. Generate a realization for $D_{v\mathbf{U},intra}$ | assigned to community $j$ ($d_{intra}$)
    5. Connect U to $d_{intra}$ subsystems inside community $j$ using the PA model
    6. For each interface established in (5), assign complexity ($w_{i\mathbf{U},intra}$)
```

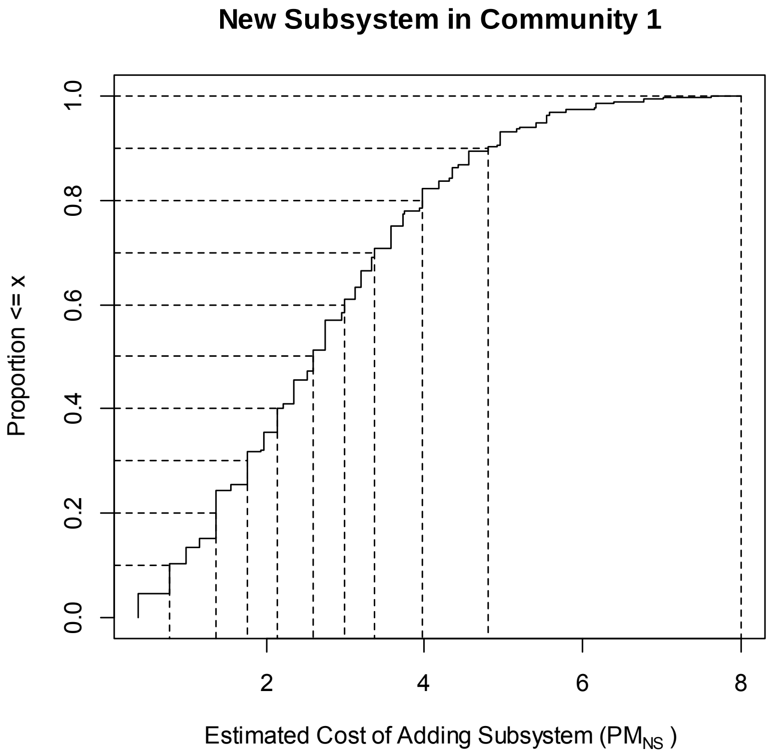
Intercommunity growth

```
    7. Generate a realization for $D_{v\mathbf{U},inter}$ | assigned to community $j$ ($d_{inter}$)
    8. Connect U to $d_{inter}$ communities using the PA model
    9. For each interface established in (8), assign complexity ($w_{i\mathbf{U},inter}$)
    10. Estimate cost for augmented system using COSYSMO ($PM_{NS}$*)
    11. Calculate additional cost of adding subsystem U ($PM_{NS}* - PM_{NS}$)
    12. Store results and return to (3)
```

Cost Insights from Monte Carlo Simulation

Adding New Subsystem to Community 1



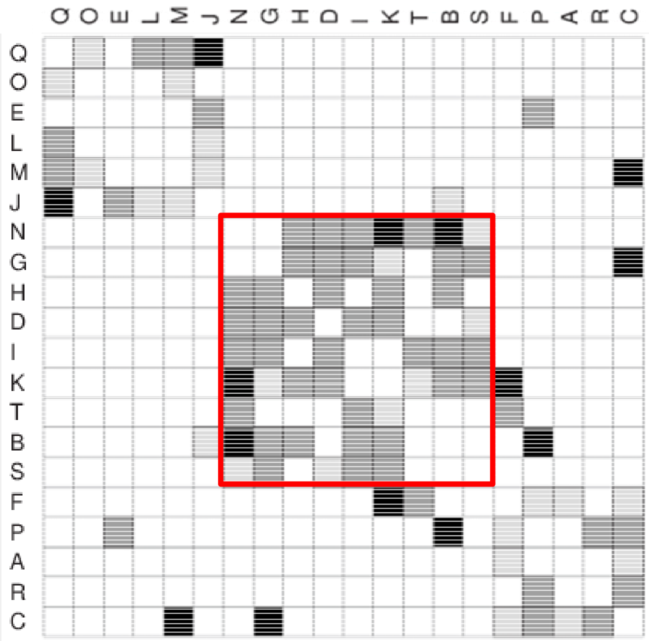

New Subsystem in Community 1

- 95% CI is (2.68, 2.78) in PM_{NS}
- “Best guess” for the cost of adding subsystem U is $54,740
- Cost to add subsystem U should not exceed $160,020

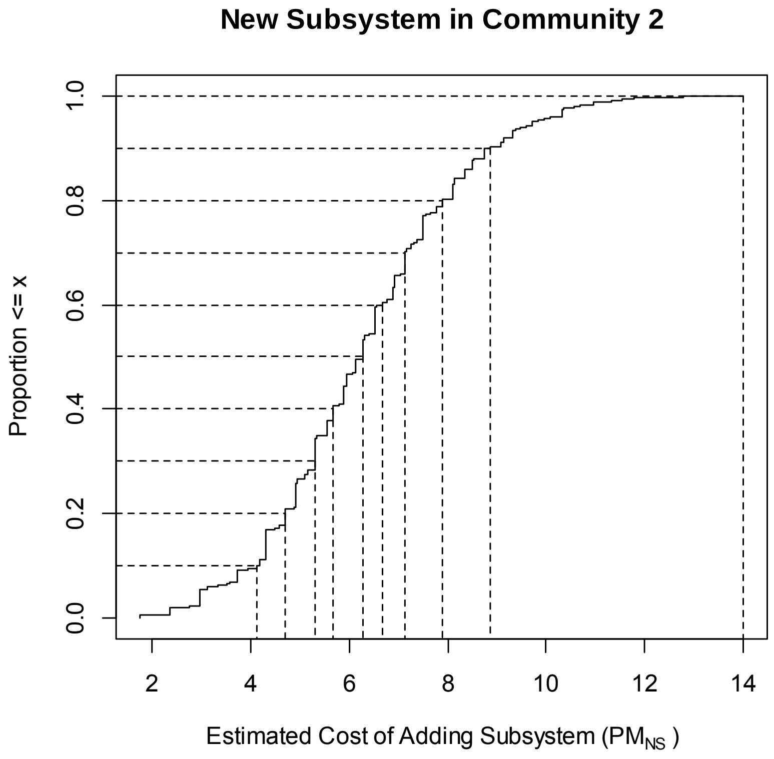
ECDF for the additional cost of adding subsystem U to Community 1

Cost Insights from Monte Carlo Simulation

Adding New Subsystem to Community 2



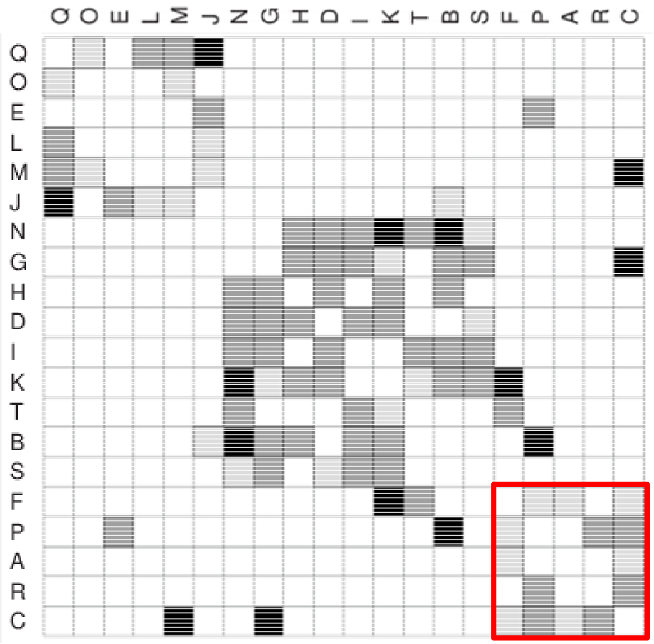
- 95% CI is (6.25, 6.39) in PM_{NS}
- “Best guess” for the cost of adding subsystem U is $126,520
- Cost to add subsystem U should not exceed $280,240



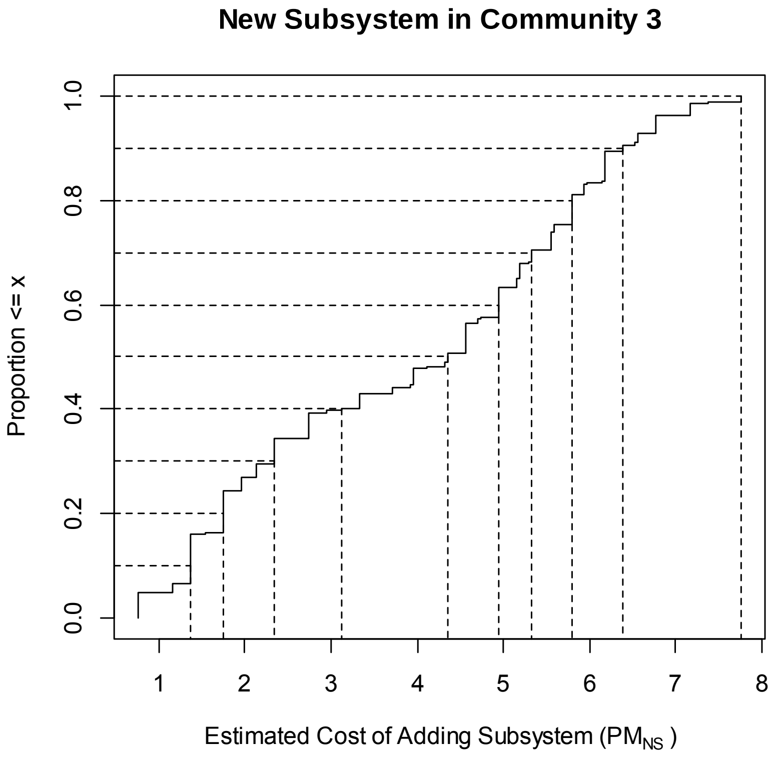
ECDF for the additional cost of adding subsystem U to Community 2

Cost Insights from Monte Carlo Simulation

Adding New Subsystem to Community 3



- 95% CI is (3.86, 4.00) in PM_{NS}
- “Best guess” for the cost of adding subsystem U is $78,720
- Cost to add subsystem U should not exceed $155,120


New Subsystem in Community 3

ECDF for the additional cost of adding subsystem U to Community 3

A Few Limitations

- Technological interfaces are not random; they are engineered based on requirements
  - Concur; however, detailed interfaces are engineered AFTER requirements mature; this is an early lifecycle estimate
- Addition of subsystem U could reasonably necessitate the “rewiring” of the existing architecture
  - Concur; should probably be treated as a higher order effect
- Using existing architecture to estimate future architecture fails to account for revolutionary change
  - Concur; however, our approach addresses evolutionary change; wholesale redesign is outside the scope of our work

Future Work

- Our immediate research efforts are . . .
  1. Conceptual Modeling
     - Adding more than 1 subsystem
     - Rewiring existing architecture
     - Accounting for “nodal properties”
  2. Exploring the relationship between COSYSMO and the remaining 24 DoDAF models required in the CDD
  3. Validation
     - Does our methodology adequately model how technical systems “grow”

Questions

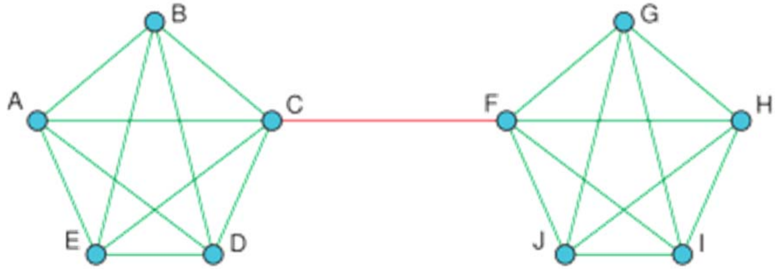
CORRESPONDING AUTHOR:
LTC Matt Dabkowski
mfd1@email.arizona.edu
matthew.dabkowski@us.army.mil

TOPIC TITLE: The Budding SV3 – Estimating the Cost of Architectural Growth Early in the Life Cycle

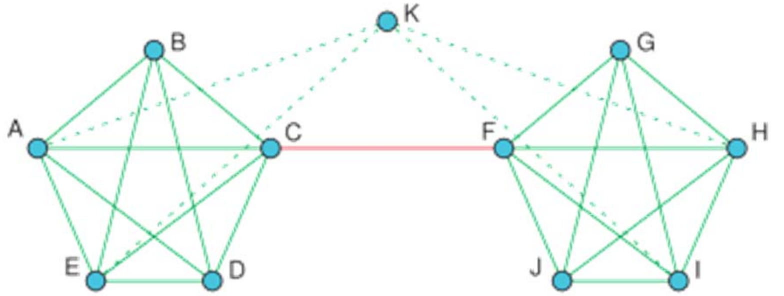
Backup Slides

Why Communities Matter - A Thought Experiment

- Start with this system . . .



- Run without Girvan-Newman . . .
- And get this . . .



<u>Existing Architecture</u>
- 2 communities
- Bridging ties rare
- Bridging ties difficult

<u>New Architecture</u>
- 2 communities?
- Bridging ties rare?
- Bridging ties difficult?

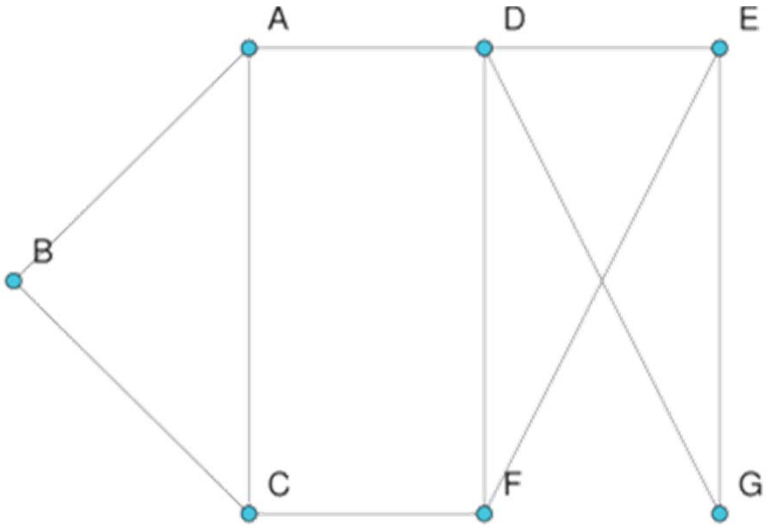
When it comes to incremental growth, existing community structure matters!

Girvan-Newman Algorithm

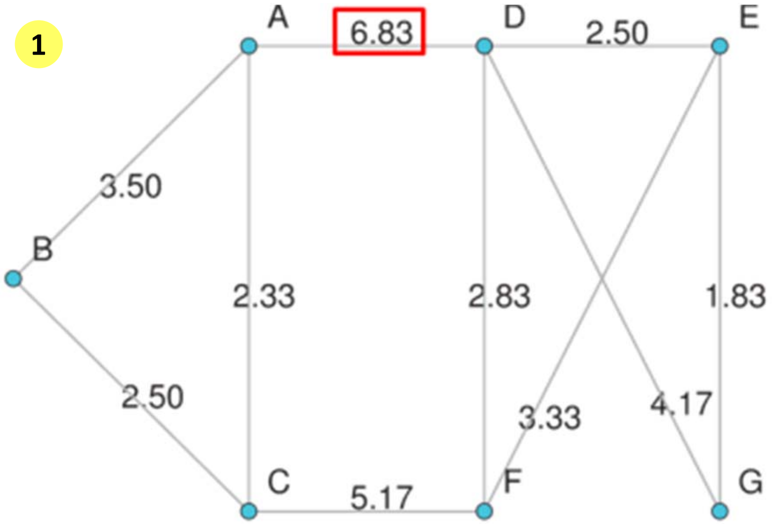
- Based on the idea of edge betweenness (*eb*), the number of geodesic (shortest) paths that contain an edge
- Edge's with high *eb* bridge communities of vertices, groups of vertices where the number of edges is dense within and sparse between groups
- Sketch of Girvan-Newman (GN)
    1. Given network *G* of size *n*, calculate *eb* for each edge in *G*
    2. Delete edge with the highest *eb* from *G*, yielding subgraph *G'*
    3. If *G'* has 0 edges, terminate; otherwise, set *G* as *G*' and return to 1
- At termination, GN produces a dendogram (tree) with *n* leaves
- Cutting the tree at different levels produces different community structures
- Community structure that maximizes modularity is selected

A Simple Example (1 of 4)

Full Graph and Edge Betweenness



Full graph (*G*) has 7 vertices and 10 edges

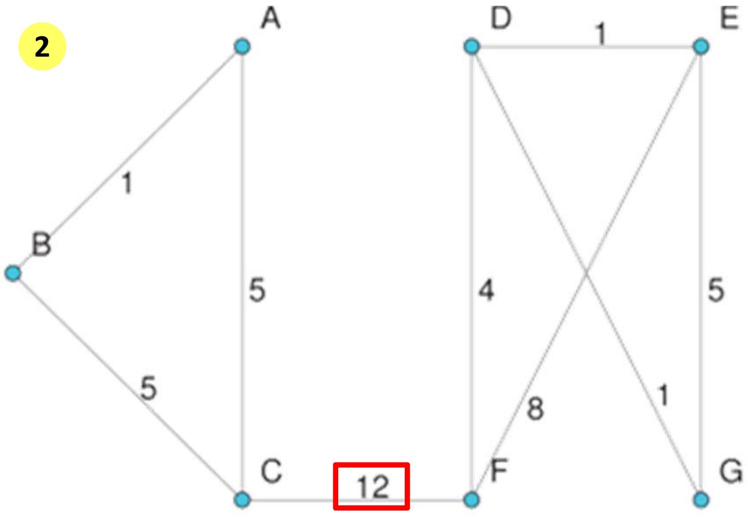


Edge betweenness (*eb*) is calculated for each edge in *G*

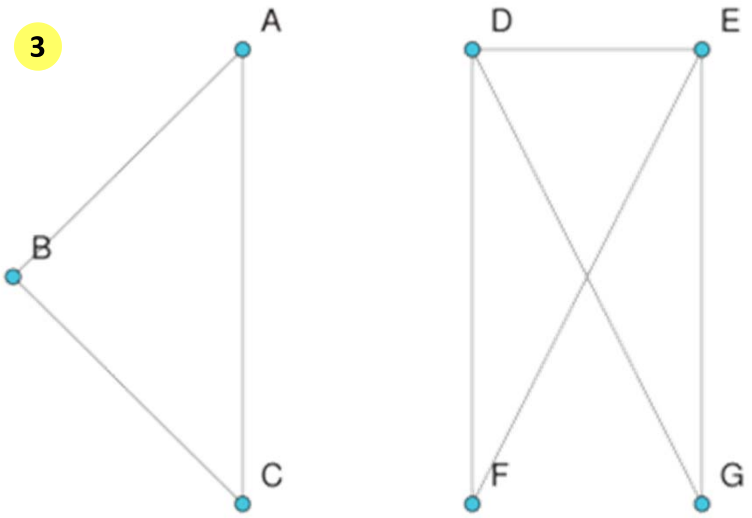
- Example edge betweenness calculation ($eb_{B-C} = 2.5$)
- B-C is the shortest path between B and C ($eb_{B-C} = 1$)
- B-C is on the unique shortest path between B and F (B-C-F → $eb_{B-C} = 2$)
- B-C is on 1 of 2 shortest paths between B and E (B-A-D-E and B-C-F-E → $eb_{B-C} = 2.5$)
- Edge A-D has the highest edge betweenness ($eb_{A-D} = 6.83$); delete A-D from *G*

A Simple Example (2 of 4)

Edge Deletion and Fragmentation



- Edge A-D is deleted from *G* leaving *G'*
- *G'* has at least one edge; *G* = *G'*
- Edge betweenness is calculated for *G*
- Edge C-F has the highest edge betweeness



- Edge C-F is deleted from *G* leaving *G'*
- *G'* has at least one edge; *G* = *G'*
- *G* is now fragmented into 2 communities

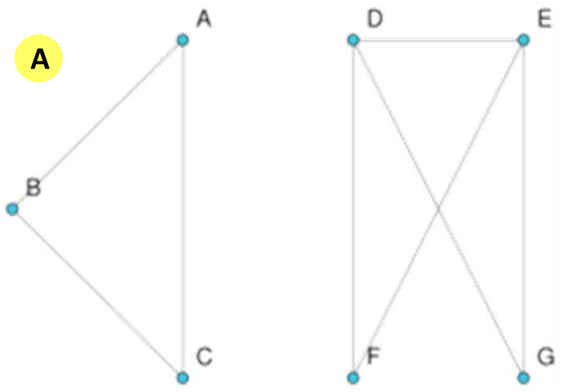
- After the deletion of edge C-F, *G* has two communities (A,B,C) and (D,E,F,G)
- How "good" is this division? Modularity provides an answer . . .

A Simple Example (3 of 4)

Modularity (Q)

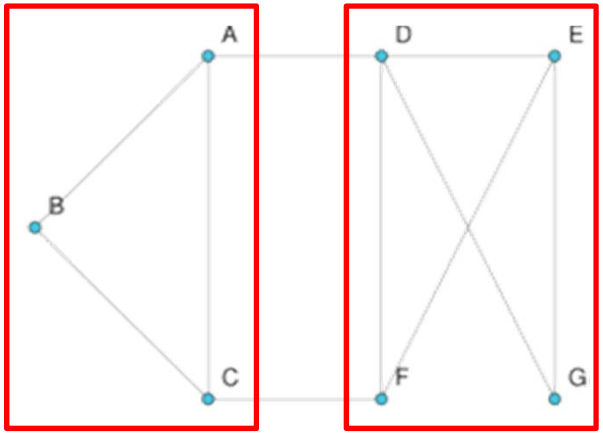
$$Q = \sum_{i=1}^{n} \left(e_{ii} - a_i^2\right)$$

A



After the removal of edge C-F,
G is fragmented into 2 (*n*) communities

B



Using the full graph, calculate e_{ij}, fraction of edges between communities *i* and *j* (for all *i, j*)

C

	1	2
1	0.3	0.1
2	0.1	0.5

Using e_{ij}, build matrix ***e***

D

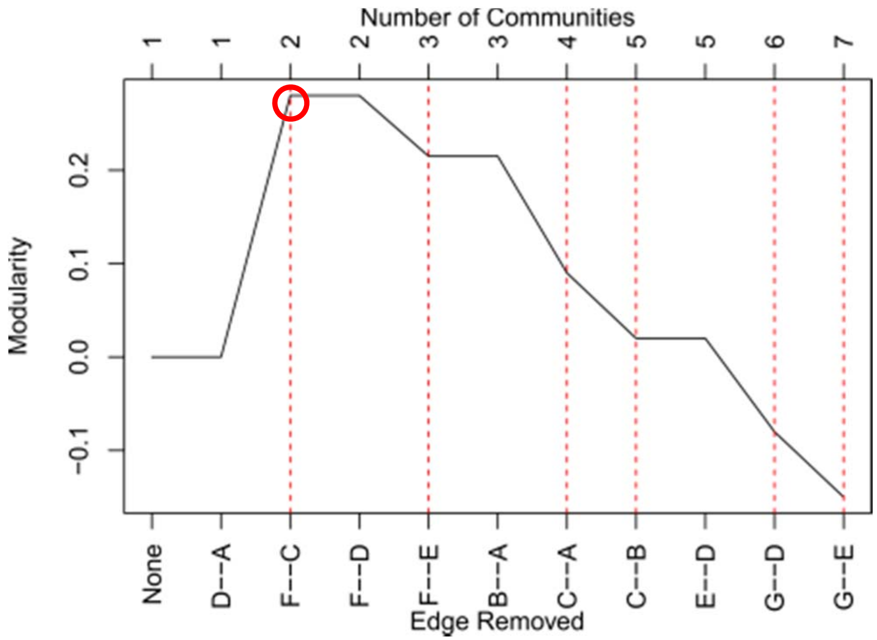
Row Sums

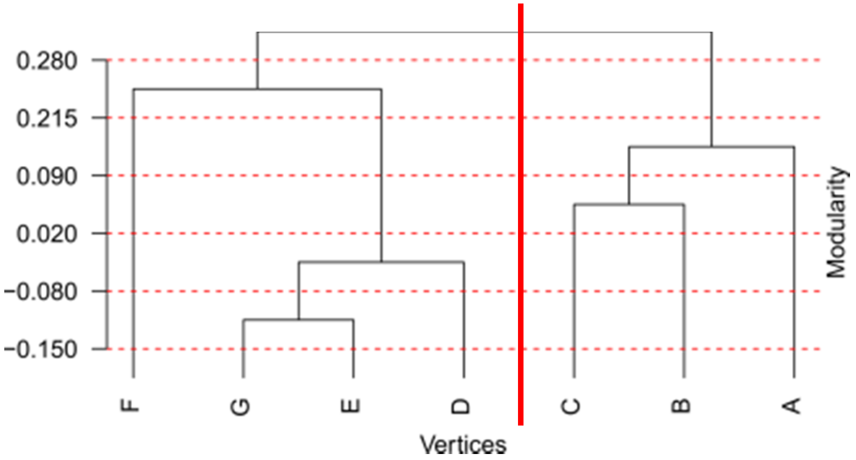
a_i	a_i^2	$e_{ii} - a_i^2$
0.4	0.16	0.14
0.6	0.36	0.14
	Q	0.28

Calculate modularity (*Q*), where a_i is the sum of row *i* of ***e***

A Simple Example (4 of 4)

Termination, Modularity Maximization, and Community Selection





Dendogram

- Process continues in a similar manner until *G* has no edges remaining
- At termination, the dendogram has 7 (*n*) leaves with 6 potential community structures
- Our initial fragmentation maximizes modularity at 0.280
- Conclusion: *G* contains 2 communities, namely, (A,B,C) and (D,E,F,G)