Faculty Researchers:
Richard Turner, Stevens Institute, PI
Alice Smith, Auburn University, Co-PI
Jeff Smith, Auburn University

Student Researchers:
Auburn University: Gokhan Ozden, Donghuang Li
Stevens Institute: Rick Scarpetti, Lymari Castro
USC: Alexey Tregubov

SERC Sponsor Research Review
November 17, 2016
• Overview and Status
• Observations from RT-126
• Examples of Refined Definitions
• Walkthrough of example
• Next steps
Agile (Adaptive) SE goals

• Identify agile, lean, and other adaptive processes and governance mechanisms to help systems engineers
  — Identify, analyze and quickly **react** to issues in an environment of accelerating change
  — Keep pace with **evolving** requirements, risks and opportunities **throughout** the extended development lifecycle
  — Understand and manage the changing **economic and political factors** that undergird and enable system development
  — Broaden SE influence and **holistically** approach
    o Creation and evolution of systems of systems
    o Interoperability between legacy and new capabilities
    o Reductionism resulting in point solutions or locally optimized decisions

• A modeling environment to validate and experiment with
  — Adaptive mechanism performance
  — Their interactions with more traditional SE
  — How they can balance adaptability with discipline in a broad variety of environments.

• Inform organizations contemplating changes to their system development processes in complex system or SoS environments where different development approaches are applied concurrently.
• Key enabler for both conduct and transition of the research

• A simulation-based research infrastructure to study, validate, and demonstrate to decision makers the costs, benefits, and risks of adaptive approaches

Objectives
Enable realistic experiments to understand how governance models, organizational structures and work flows interact across a SoS
- Investigate new approaches in a wide variety of acquisition and development environments
- Understand the implications of organizations with mixed governance models across components
- A possible means of measuring the value of systems engineering

Provide a framework to calibrate assumptions of performance
- Build data for increasingly sophisticated experiments
- Resource capabilities, team maturity, turnover
- Overhead costs, complexity of work, rate of environmental change

Integrated experiment-generation tools that provide the user with
- A broad selection of calibrated and experimental patterns for organizations and governance models
- Sophisticated stochastic support for generating large work item networks
- Comparison to other similar experiments
- Graphic demonstration of benefit/cost to convince risk-averse decision makers
**Domain Specific Language**: Define environmental and management characteristics

- **Organizations**: Captures various organizational structures, including SoS
- **Work flows**: Defines the technical work to be accomplished
- **Governance**: Specifies how work is accomplished across and within organizations

**Implementation infrastructure**: Software libraries that implement the DSL models for simulation

- Enablers for executing the DSL syntax and semantics

**Simulation engine(s)**: Software that performs and controls the simulations

- Initially using RePast an agent-based simulator
- Research should generate sufficient data to add other simulation modes or specialized simulations such as systems dynamics models

R-126 delivered an initial prototype of DATASEM in a web-accessible and standalone edition in December 2015. However, ...
• Initial prototype was completed and delivered to the sponsor in December 2015

• Start of 2016 work delayed until March

of the prototype by the team led to a number of observations and the decision to address some foundational issues

• Have revised the definitions of many of the modeling structures
  — Currently undergoing review
  — Revising the DSL to align with the changes
  — Beginning revision of the software
Observations from RT-126 (-) and Responses in RT-152 (+)

- Vague definitions of governance strategies and mechanisms
  - The definitions were misunderstood and so implemented incorrectly
  + Definitions have been updated and clarified
  + New definitions being reviewed by industry and sponsors
  + Building a set of real world scenarios showing how mechanisms work

- Interaction of organization, work and governance models and work flow definition
  - Had assumed (incorrectly) that the models were independent
  + New definitions and templates
  + Incorporate interaction

- DSL
  - Tension between elegance, extensibility, and practicality
  + DSL being revised based on the new definitions
Refined Definitions
Organizational Components (OCs)

OCs are companies, organizations, teams, individuals, resources – are defined according to their capabilities, governance models and relationships in a data-rich graph.

Each OC has a requested work queue and an accepted work queue.

Each OC has 4 key functions:
- **Accept work**
- **allocate Resources**
- **Execute work items,**
- **Monitor and report status.**

Each OC has a **Governance Model,** defines the scope, concurrence and activity of the technical process and management mechanisms.

**Two types of OCs**

**Basic.** Basic OCs have only internal resources. Can represent tools or special equipment (for example a test suite configuration), or individual staff resources.

**Delegating.** Delegating OCs perform work using internal resources or by arranging to delegate work with external OCs of any type.

**Two types of relationships in the OCs graph:**

**Workflow relationship.** A relationship that defines how requests for work can flow between organizational components. The directionality and parameters of the relationship are defined in the individual OC’s governance model.

**Management hierarchy.** Identifies where there are hierarchical or management relations. Every OC operates according to a governance model. Unless a governance model component is specifically defined for an OC, the OC inherits the component from its parent in the hierarchy.
Definitions

Example Organization Graphs

Single-team Organization

Multiteam Organization

These two teams can either delegate work to lower level (downstream flow) or execute it internally.

These are basic OCs. They accept work and execute it internally and do not delegate accepted work to other OCs.

Red arrows show work delegation (downstream) and blue arrows show status monitoring or WI status updates (upstream).

Basic OCs can delegate work back to upstream OC for redistribution only if it is a new WI that was triggered by the current WI under execution (e.g. request for additional analysis, unforeseen work, etc.).
- Square boxes show artifacts or data structures
- Diamonds show agents’ governance mechanisms components or activities/functions
Because there is a relationship between the technical work decomposition and scheduling mechanisms, two types of scheduling approaches were defined:

This simple binary choice is an initial representation of an extended spectrum.

Using the PMI definitions, we have described the approaches as:

- **Predictive.** Work is scheduled far ahead of time, batch size of tasks is usually large, progress is based on that schedule, and the schedule is adjusted rarely.
- **Adaptive.** Work is scheduled continually or in short iterations, batch sizes are usually smaller, value may be used to prioritize work, and flow controls may be used.
Acceptance chooses what and how many WIs to take from the external queue into the accepted queue. Acceptance can either be triggered by an event or by cadence.

The acceptance process has the following set of parameters for adaptive OCs:

**Cadence** - how often acceptance is performed

**Scheduling Mechanism** - an instance of a scheduling mechanism (e.g. schedule, sprint, kanban-pull, ...)

**Value Determination** - this mechanism uses user-specified parameters (e.g. weights of different criteria, method of aggregation of different criteria, etc.) to determine the value of work items in the queue

**Work Flow Mechanisms** – CoSs, WIP limits.
The acceptance process has the following set of parameters for *predictive* OCs:

- **Cadence**
- **Acceptance policy** towards new work in the external queue.

Currently there are three policies for acceptance:

1. Accept everything from external queue and assign to resources in parallel to already existing plan. First try to utilize available resources, and only then multitask busy resources.
2. Reschedule everything every time something new appears in the external queue.
3. Ignore external queue between rescheduling.

No matter what acceptance policy is used, rescheduling is also done with a certain cadence (every month, year, etc.). Rescheduling cadence is a user-defined parameter.
• Described by Work Items (WIs) in a WI Network (WIN) with multiple levels resembling a traditional work breakdown structure (WBS).

• Users specify information for the first few levels of work (e.g. capabilities, requirements, work packages); the rest of the network is stochastically generated using parameters OC technical approach and governance models.

• Stochastic work item decomposition is performed when the WI is accepted by an OC. A Work Decomposition task is created. Once that task is complete, a WIN-generation algorithm creates a new set of nodes using parameters defined in the WI and the OC’s governance specification.
Example (using the new definitions)

• Organizational structure

*FTE – full time equivalent
Example of using the new definitions (cont-d)

Process-independent work definition (blue WIs on the next chart) includes three capabilities, their decomposition into requirements and project level requirements. The SE team delegates project level requirements to OCs that decompose them into task level WIs (red WIs on the next chart). The work decomposition steps:

- Capabilities are allocated to SE team. Simulation engine allocates capabilities (roots of the WIs graph) based on required skills.

- Capabilities are decomposed into requirements by SE team.

- Requirements are decomposed into project level requirements by SE team.

- SE team delegates PRs to other team, other teams decompose PRs into tasks using their governance settings
Example of using the new definitions (cont-d)

Assigned / delegated by simulator

PA01 → C1
PA02 → A01
PA03 → A02
PA04 → A03
PA05 → A04
PA06 → A05
PA07 → A06
PA08 → A07

C1 → R1
C1 → R2
C1 → R3
C1 → R4

R1 → A01
R1 → A02
R1 → A03
R1 → A04
R1 → A05
R1 → A06
R1 → A07

R2 → A01
R2 → A02
R2 → A03
R2 → A04
R2 → A05
R2 → A06
R2 → A07

R3 → A01
R3 → A02
R3 → A03
R3 → A04
R3 → A05
R3 → A06
R3 → A07

R4 → A01
R4 → A02
R4 → A03
R4 → A04
R4 → A05
R4 → A06
R4 → A07

PR1 → R1
PR2 → R2
PR3 → R3
PR4 → R4

PR1 → PR2
PR2 → PR3
PR3 → PR4
PR4 → PR5
PR5 → PR6

Inheritance – decomposition hierarchy
Precedence (source precedes/cause destination)

Team 1
2 FTE
Skills: s1, s2 adaptive

Team 2
4 FTE
Skills: s2 adaptive

Team 3
2 FTE
Skills: s3 predictive

Process independent WIs
(input from user, from DSL)

WIs introduced by simulation

Tasks generated by OCs
Decomposition tasks created by OCs. This is governance overhead.
Planning tasks created by OCs. This is governance overhead.

GOPRINT PACKAGE

Generated by (source task creates destination tasks and/or relationships). A source tasks is a decomposition task.

Generated by (source task creates destination tasks, but not schedules them). Only for predictive processes.

Inheritance – decomposition hierarchy
Precedence (source precedes/cause destination).
Example Process-independent work definition

<table>
<thead>
<tr>
<th></th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>PR1</th>
<th>PR2</th>
<th>PR3</th>
<th>PR4</th>
<th>PR5</th>
<th>PR6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>40</td>
<td>30</td>
<td>30</td>
<td>20</td>
<td>35</td>
<td>30</td>
<td>15</td>
<td>10</td>
<td>10</td>
<td>35</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>Value distribution for</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>constituent WIs.</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Distribute value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>proportionally (divide</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>parent’s value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>proportionally).</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Estimated size</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Estimated error</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Required skills</td>
<td>S0</td>
<td>S0</td>
<td>S0</td>
<td>S0S</td>
<td>S0</td>
<td>S0</td>
<td>S0</td>
<td>S0</td>
<td>S0</td>
<td>S0</td>
<td>S0</td>
<td>S0</td>
<td>S1</td>
</tr>
<tr>
<td></td>
<td>S1</td>
<td>S2</td>
<td>S2</td>
<td>S2</td>
<td>S2</td>
<td>S2</td>
<td>S2</td>
<td>S2</td>
<td>S2</td>
<td>S2</td>
<td>S2</td>
<td>S2</td>
<td>S2</td>
</tr>
<tr>
<td></td>
<td>S3</td>
<td>S3</td>
<td>S3</td>
<td>S3</td>
<td>S3</td>
<td>S3</td>
<td>S3</td>
<td>S3</td>
<td>S3</td>
<td>S3</td>
<td>S3</td>
<td>S3</td>
<td>S3</td>
</tr>
<tr>
<td>Distribution of skills</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>for constituent WIs</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1-100%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2-100%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2-100%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2-100%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3-100%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3-100%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Probability of change</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Constituent WIs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>internal relationships</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>complexity</td>
<td>0-90%</td>
<td>0-90%</td>
<td>0-90%</td>
<td>0-90%</td>
<td>0-90%</td>
<td>0-90%</td>
<td>0-90%</td>
<td>0-90%</td>
<td>0-90%</td>
<td>0-90%</td>
<td>0-90%</td>
<td>0-90%</td>
<td>0-90%</td>
</tr>
<tr>
<td>1-10%</td>
<td>1-10%</td>
<td>1-10%</td>
<td>1-10%</td>
<td>1-10%</td>
<td>1-10%</td>
<td>1-10%</td>
<td>1-10%</td>
<td>1-10%</td>
<td>1-10%</td>
<td>1-10%</td>
<td>1-10%</td>
<td>1-10%</td>
<td>1-10%</td>
</tr>
<tr>
<td>Constituent WIs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>external relationships</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>complexity</td>
<td>0-100%</td>
<td>0-100%</td>
<td>0-100%</td>
<td>0-100%</td>
<td>0-100%</td>
<td>0-100%</td>
<td>0-100%</td>
<td>0-100%</td>
<td>0-100%</td>
<td>0-100%</td>
<td>0-100%</td>
<td>0-100%</td>
<td>0-100%</td>
</tr>
<tr>
<td>Deadline (days)</td>
<td>8</td>
<td>12</td>
<td>20</td>
<td>8</td>
<td>8</td>
<td>12</td>
<td>20</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>12</td>
<td>12</td>
<td>20</td>
</tr>
</tbody>
</table>
Use Scenario Example (still being defined)

• **Title:** Systems of Systems

• **Description:** A Systems of Systems project provides opportunities for a good mix of work to be completed in house and work to be contracted out, which is often not an easy decision for a company to make.

• **Potential Considerations:** Work item value can potentially serve a dual purpose, the first being the traditional sense of value delivered to customer, but the work items may have intrinsic value to the OC performing the work if it helps develop/maintain domain knowledge related to a core value stream. It may not be in the company’s long term interest to delegate WI’s related to a value stream even if it enhances value delivered in the short term.

Rick Scarpetti’s ongoing work...
Future work

• **Calibration and validation.** DATASEM needs to be validated through rigorous experimentation for sensitivity, and through the use of real data from development organizations for improved accuracy and realism. An experimental validation framework was created by Forrest Shull from the SEI.
  
  — Validation of tool in development and acquisition environments (ongoing)

• **Concepts still to be implemented or further studied**
  
  — Value determination and agreement (ongoing)
  
  — Interface of predictive and adaptive governance mechanisms (beginning)
  
  — Extend DATASEM to address acquisition (ongoing)
  
  — Model human behavior in decision making and negotiation among components of a system of systems (Not yet begin)
  
  — Conflicts between SoSE and constituent systems (not yet begun)
  
  — Negotiation of services and contracts (not yet begun)
  
  — Incorporate DATASEM into the SEEA (not yet begun)

• **Develop user interface for non-developers.** DATASEM’s UI enhanced so a typical systems engineering or engineering manager can create products readily understandable by decision makers and executives.
  
  — Simplified graphic design and display interface (beginning)
Contact information:
Rich Turner
rturner@stevens.edu
202-390-3772
Backups
Illustration of Components

- OrgComp Name
- ExtQ
- AccQ
- %CPCTY
- WIP (E)
- %OVHD
- Velocity

- Capability Name % Complete
- TWI ###
- TVal ####
- Velocity ### Avg.

- Reqt ### % Capacity Avg. 35%
- Tval ### Velocity Avg. 32
- TWI ###
Refined Definition

Value Function Mechanism (Example)

Different prioritization criteria:
- Value criteria: “Inherited value”
- Value criteria: “Estimated size”
- Value criteria: “Arrival time”
- Value criteria: “Number of dependent WIs”
- Value criteria: “Resource availability”
- Value criteria: “…”

Aggregate Value function
Parameters:
- criteria 1 weight
- criteria 2 weight

Example of aggregating value function:
Weighted sum of all components (by order) – each WI is ordered by all criteria and then weighted sum of orders is computed.

<table>
<thead>
<tr>
<th>WI id</th>
<th>Prioritization component algorithm “Inherited value”</th>
<th>Prioritization component algorithm “Estimated size”</th>
<th>Prioritization component algorithm “Arrival time”</th>
<th>Prioritization component algorithm “Number of dependent WIs”</th>
<th>Prioritization component algorithm “Resource availability”</th>
<th>Aggregated</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>9</td>
<td>5</td>
<td>1</td>
<td>10</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>10</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>
**Adaptive** technical processes often use multiple iterations or increments with constant or changing cadences to provide continuous value and validation. They may order WIs according to some value formulation, or handle resource constraints due to the changing environment of the project. They may enforce broad Classes of Service that are used by other OCs, internal Work In Process limits, or other methods to support better flow.
Predictive technical approaches usually generate (or obtain) a schedule and then execute it as closely as possible, redeveloping schedules when they reach some defined level of incongruence to provide insight and make management adjustments.
More Definitions...

**Monitoring**

1. For each accepted WI do:
2. Update all higher level WIs by recalculating their properties (e.g., completeness, status, etc.)
3. No
4. Update output indicators

**Internal Work Execution**

1. For each WI assigned to internal execution do:
2. Update WI completeness using resource performance parameters (capacity, throughput).
3. Develop a new schedule using new and existing WIs and current schedule if available.
4. Generate WIN for the associated aggregating WI
5. Add new WIs into accepted queue

Any WI that generates other WIs (e.g., analysis tasks, rescheduling tasks, etc.). This is determined by type of the WI.

Every analysis/rescheduling task has a parent, which is an aggregating WI.
Publications from this research-1
(in order of publication)


• Turner, R; L. Yilmaz; J. Smith; D. Li; S. Chada; A. Smith; A. Tregubov. “DATASEM: A simulation suite for SoSE management research,” 11th System of Systems Engineering Conference (SoSE), Kongsberg, Norway, June 2016.