

# Ricks 19755

# Agile Dynamics at Scale

## A MITRE Innovation Program Research Project

---

NDIA 20<sup>th</sup> Annual Systems Engineering  
Conference

Presenting author:

Aleksandra Markina-Khusid [amk@mitre.org](mailto:amk@mitre.org)

# Outline

---

- **Project Description**
- **Modeling Agile Dynamics at Scale**
- **Simulating a Real Project**

## **Acknowledgement**

**Disciplined Agile copyright material used with permission. All rights reserved.**

**Managed Agile Delivery copyright material used with permission. All rights reserved.**

**Scaled Agile Framework copyright material used with permission. All rights reserved.**

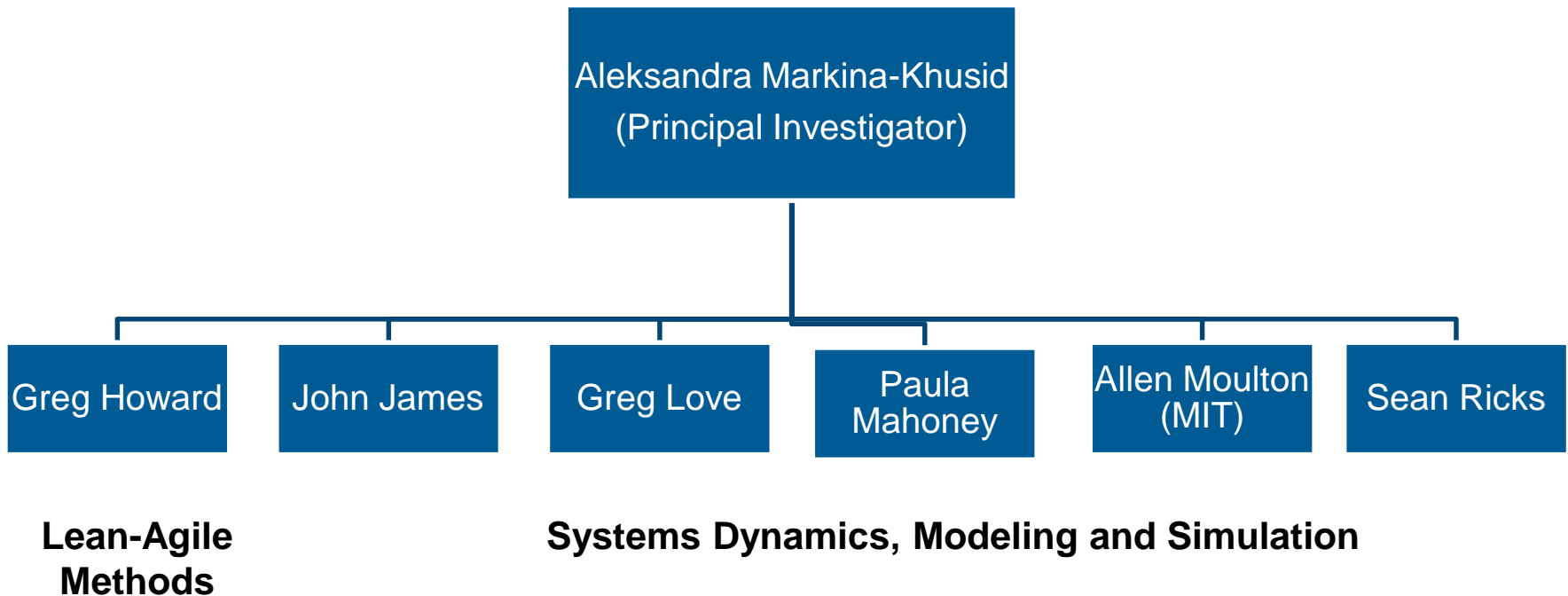
---

# Project Description

---

# Team Members

## A joint MITRE-MIT Research Project



# Goals

---

- 1. Use modeling to study how scaled Lean-Agile methods would enable Agile software development to integrate into a heavily plan-driven and risk averse enterprise such as the Air Force and DOD.**
- 2. Perform virtual experimentation with scaled Lean-Agile methods by capturing those methods in a model (or models).**
- 3. Provide expanded knowledge about Lean-Agile and a virtual experimentation resource for use by MITRE staff in engagements.**
- 4. Develop a baseline for a model that can enable MITRE staff to test alternative management structures on projects they support.**
- 5. Build a model that can make relative projections, not precise predictions.**
  - The models built in segments to test hypotheses but with a plan for integration at a later point. Each segment will provide value and contribute to Goal #1.

# Perspective User Stories

## ■ Program Systems Engineer

*Systems engineers use models to define, understand, communicate, assess, interpret, and accept the project scope; to produce technical documentation and other artifacts; and to maintain “ground truth” about the system(s).*

- DoD Acquisition Modeling And Simulation Working Group

- As a Program Systems Engineer I need to understand the engineering variables\* and trades in order to develop the Program’s Systems Engineering Plan (SEP).
- As a Program Systems Engineer and given a SEP, I need to identify risk and opportunities.

### *\*The Agile Genome*

1. Story/feature driven
2. Iterative-Incremental
3. Refactoring
4. Micro-Optimizing
5. Customer Involvement
6. Team Dynamics
7. Continuous Integration

## ■ Acquisition and Program Manager

- As a Program Manager I need to understand the SE variables impact on cost (development cost curve).
- As a Program Manager I need to understand the SE variables impact on schedule (backlog burn down and project end).
- As a Program Manager I need to understand the SE variables impact on performance (defect rate).
- As a Program Manager I need to understand the impact on cost, schedule and performance when introducing new technology into the agile development cycle.

# Research Idea

## Decision Support for Acquisition Professionals and Managers

- **Model the dynamics of Lean-Agile methods for large scale efforts on:**

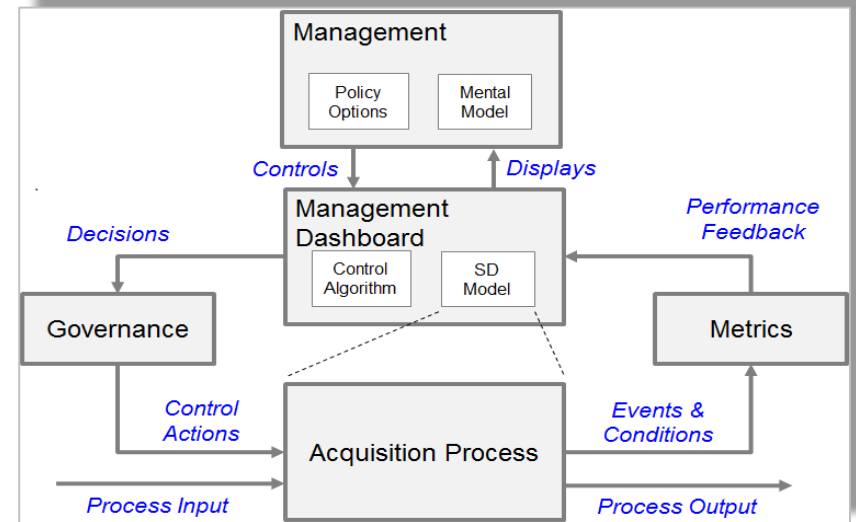
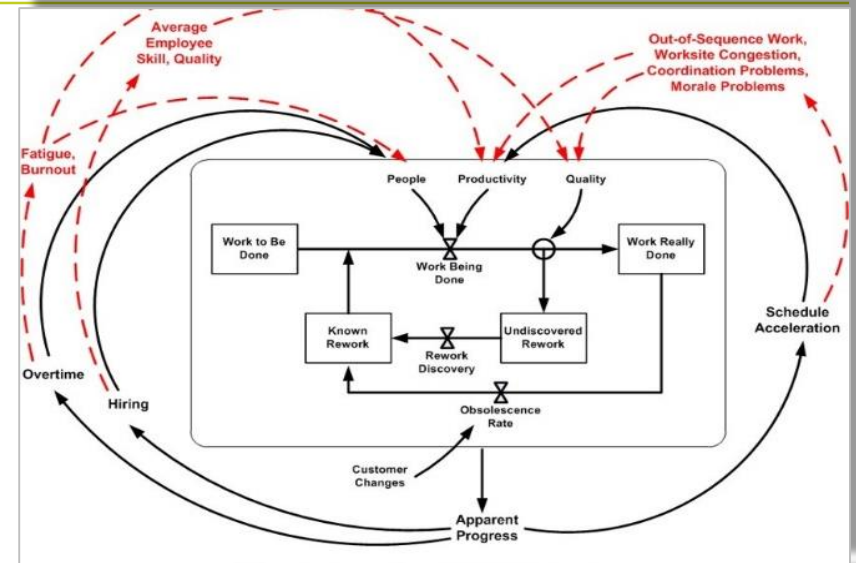
- Program acquisition
- Project management
- Systems development

- **Incorporate range of structural cause-and-effect feedback loops and factors that drive nonlinear project behaviors that impact:**

- Cost, Schedule, Performance
- Risk
- Value delivery

- **Provide dashboard tools:**

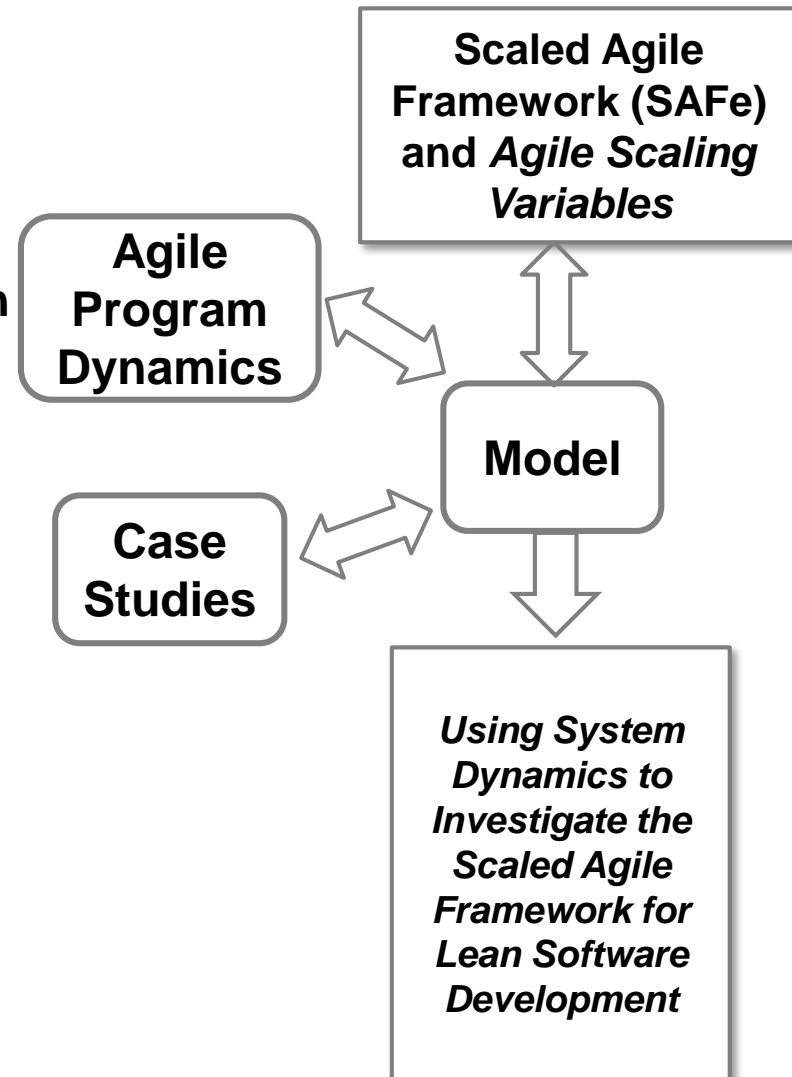
- Predictive analytics for acquisition outcomes
- Exploration of policy and governance options



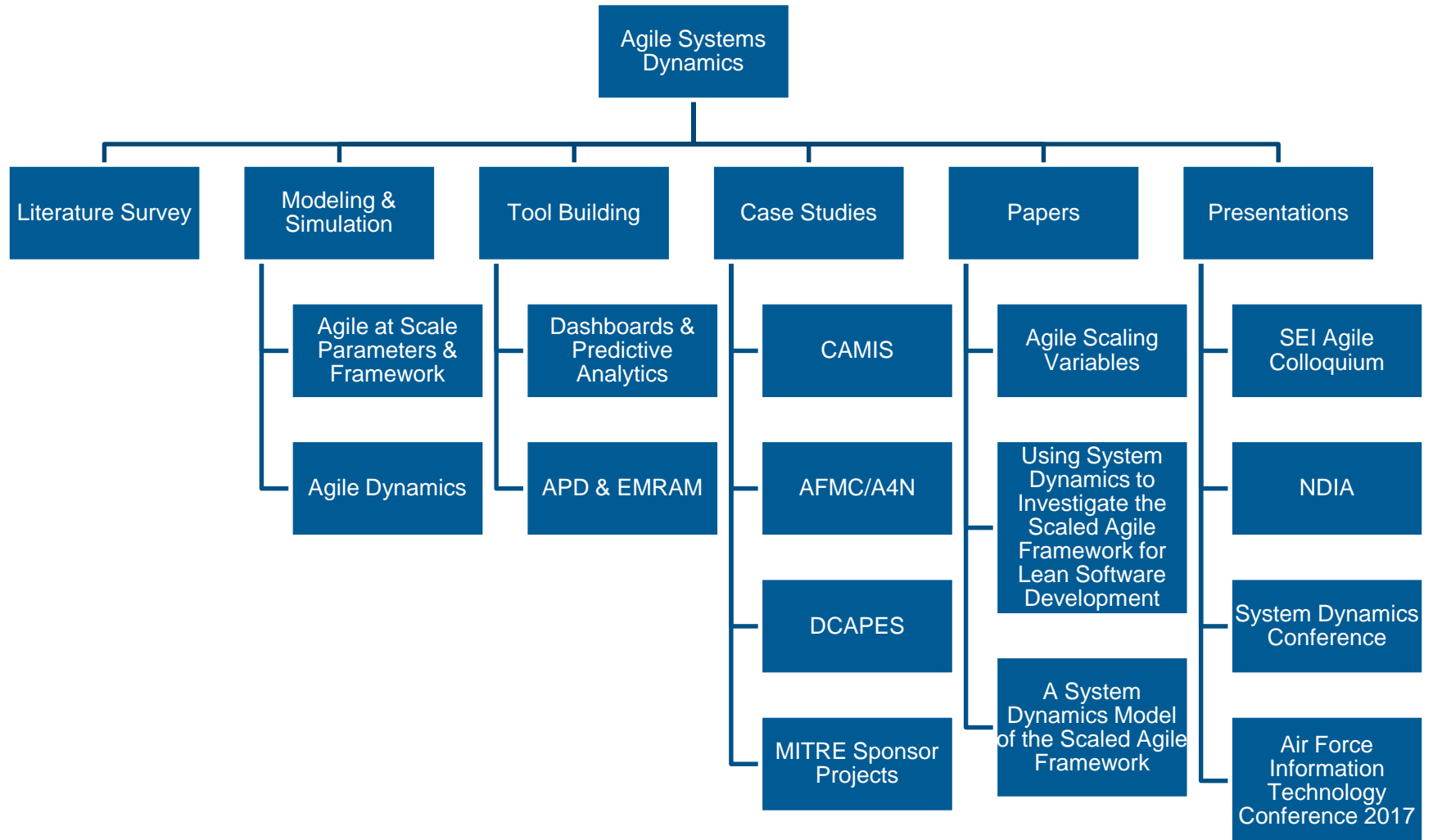


# Research Methodology

- **Builds on MIT Agile Program Dynamics model (APD)**
  - Modeled an Agile Team
  - Models **Undiscovered Rework** – a decline in quality not immediately recognized that eventually adds to **Known Work**
- **Adding SAFe and the *Agile Scaling Variables* representing Lean-Agile principles, methods and practices.**
- **Model is validated/updated with case study real world results**
  - Case studies provide and highlight the areas of modeling
- **Show that adjusting variables produce expected effects**
  - Find unexpected behavior
- **Model provides source for conference papers**



# Project Structure



---

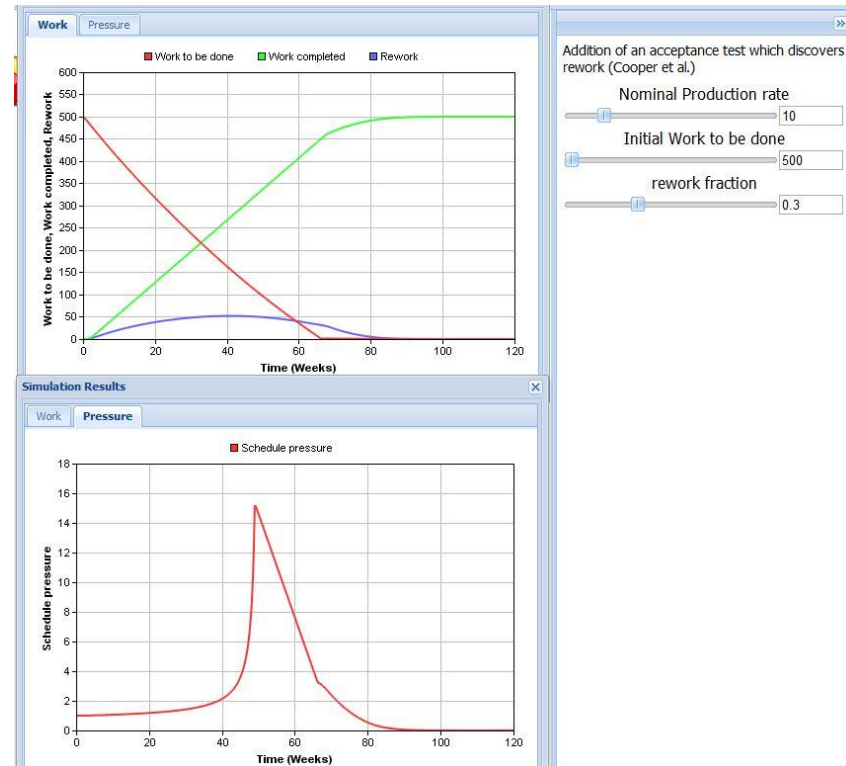
# Modeling Agile Dynamics at Scale

---

# Purpose

## ■ The rate of work completion depends on...

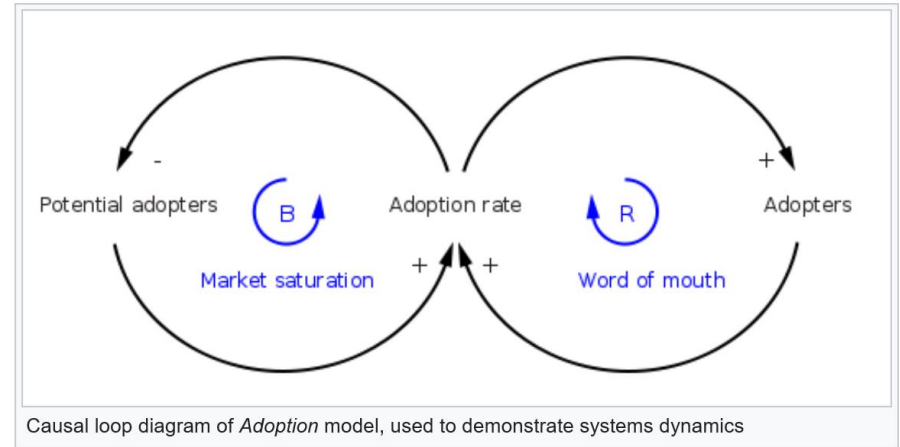
- Team size
- Number of teams
- Team experience
- Sprint duration
- Number of sprints per Program Increment (PI)
- Automated testing
- Frequency of demos
- Continuous Integration (CI)
- Etc.



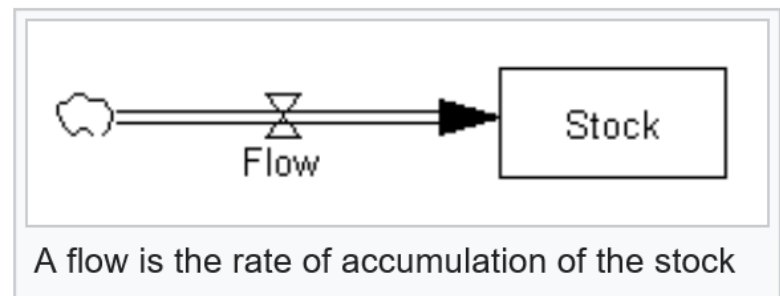
**Provide a tool to identify important dynamic relationships and trends and facilitate a conversation on process improvement.**

# Systems Dynamics

- A method to understand the dynamic behavior of complex systems
- A system's behavior is determined by:
  - Individual components, and
  - *The many circular, interlocking, sometimes time-delayed relationships among components*



The causal loop diagram visualizes how different variables in a system are interrelated



Source: Wikipedia

# System Dynamics

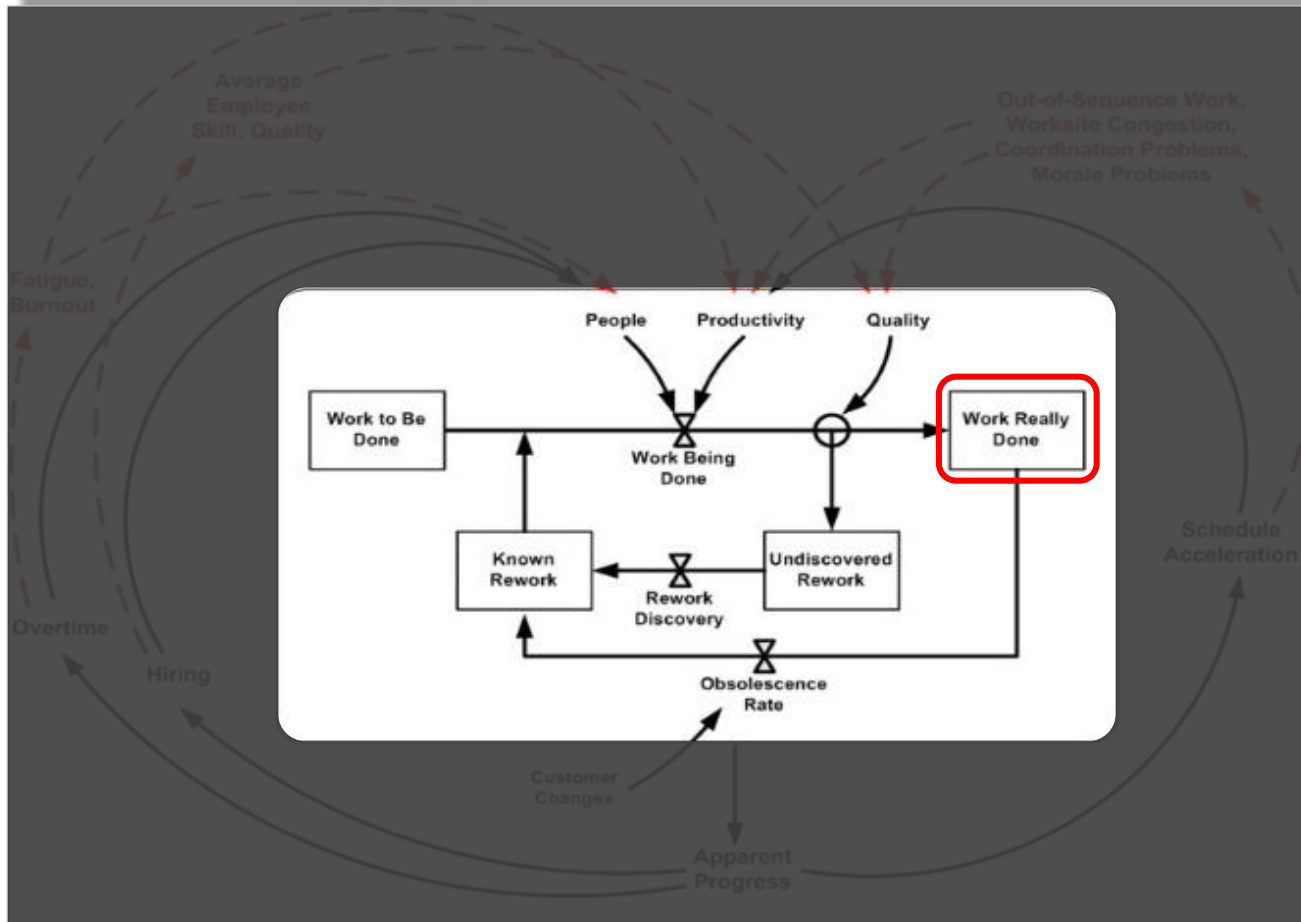
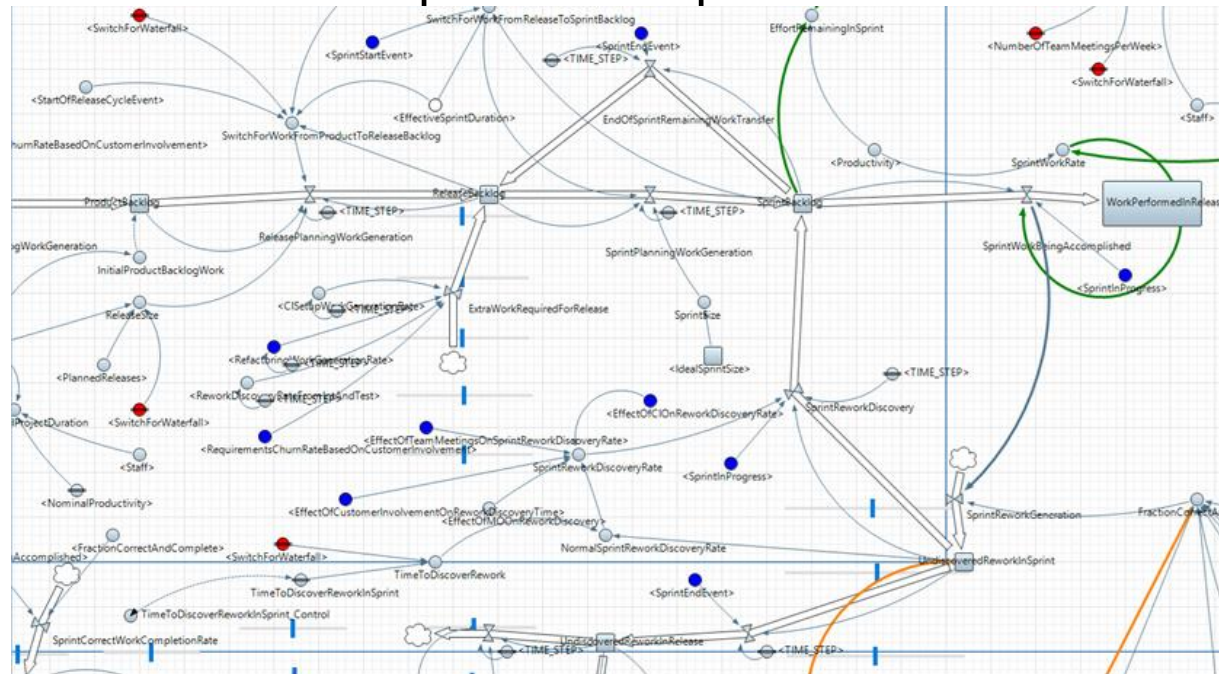


Figure 2 - Some Sample Project Dynamics

Source: Wikipedia

# Prior Work

- **Agile Project Dynamics:**
  - MIT effort, Firas Glaiel
  - Model of a single agile development team
  - Product > Release > Sprint > Completed



Glaiel, F. (2012). Agile Project Dynamics: A Strategic Project Management Approach to the Study of Large-Scale Software Development Using System Dynamics. Unpublished MIT SDM Thesis. Working Paper CISL# 2012-05.

# Prior Work

AgileProject\_v35 : Simulation - AnyLogic Professional [EVALUATION USE ONLY]
x1
experime...
AnyLogic

## AgileProject

### Model Parameters

#### Continuous Integration Sub-model

Configuration Management and Build Environment (0= no integration ... 10 = fully integrated and automated) 0 10 2

Level of Test Automation (0= no automation ... 10 = full regression test automation with nightly build) 0 10 2

Number involved in setting up the continuous integration environment (typically a small team of 1 to 3 individuals) 0 5 2

Number of weeks to setup CI environment/infrastructure 0 20 8

#### Customer Involvement Sub-model

Maximum effect of customer involvement on rework discovery time 0 1 0.5

The maximum amount by which uncertainty in customer requirements affects current fraction correct and complete 0 1 0.85

Sensitivity for the Effect of Customer Involvement on Requirements Churn 0 1 0.1

#### Refactoring Sub-model

Refactoring Aggressiveness 1 10 2

Tech Debt Accrued per Unit of Work (assumes 1 unit of technical debt per 100 tasks) 0 1 0.05

#### Release Timing Sub-model

Number of Software Releases in Agile Project 0 10 4

Release Planning Duration (in weeks) 0 5 0.5

#### Software and Integration Test Cycle Sub-model

Nominal Integration and Test Productivity (The average number of rework tasks/week) 0 20 4

Number of Integration and Test Engineers 0 20 10

### Model Controls

Use Waterfall  
 Allow Continuous Integration  
 Allow Customer Involvement  
 Allow Micro-Optimization  
 Allow Pair Programming  
 Allow Refactoring  
 Allow Integration and Test Activities

Run

### Software Development Cycle Sub-model

Maximum work intensity (assumes the ability to work 50% extra in overtime) 1 2 1.5

Nominal fraction correct and complete (FCC) 0 1 0.8

Initial Project Size 200000

Time for pressure to effect FCC (How many weeks of overtime before there is an effect of FCC) 0 10 1

Time to Discover Rework in Sprint 0 10 0.5

### Sprint Timing Sub-model

Normal Sprint planning duration (in days) 1

Sprint duration (in weeks) 4

### Staffing and Experience Sub-model

Initial Experienced Staff 10

Initial inexperienced Staff 10

Nominal time to gain experience using waterfall (in weeks) 0 25 12

Percentage of the team that changes per sprint (fraction) 0 1 0.05

Relative experience of new staff compared to old staff 0 1 0.2

### Team Dynamics Sub-model

Nominal effect of paired programming on FCC 0 1 0.5

Number of team meetings per week 0 10 5

Percent of time developers spend pair programming 0 1 0.5

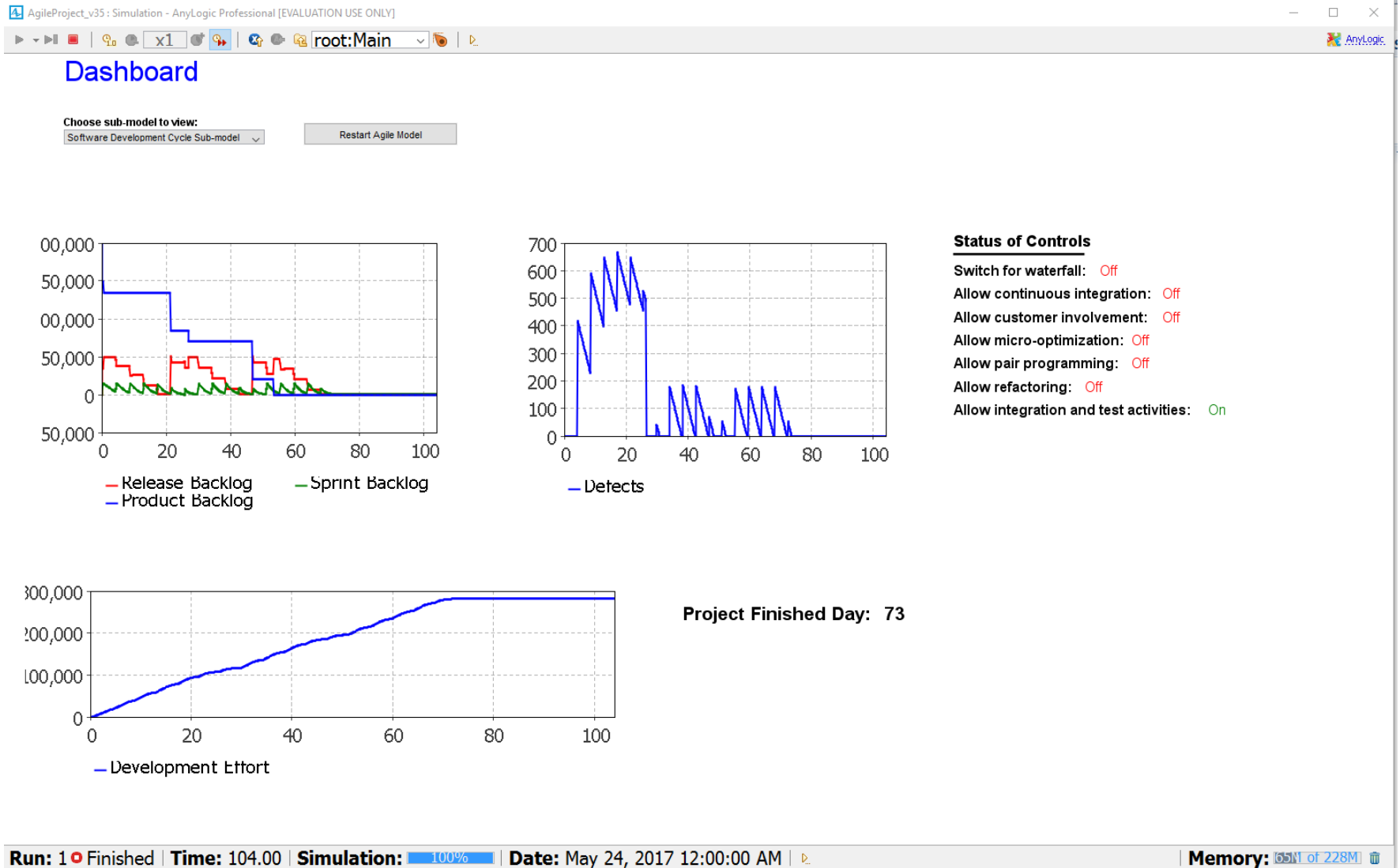
### Productivity Sub-model

Nominal productivity (tasks per person-week) 200

Run: 1 Idle | Time: - | Simulation: Stop time not set | Date: - | Memory: 4.1 M of 228M



# Prior Work

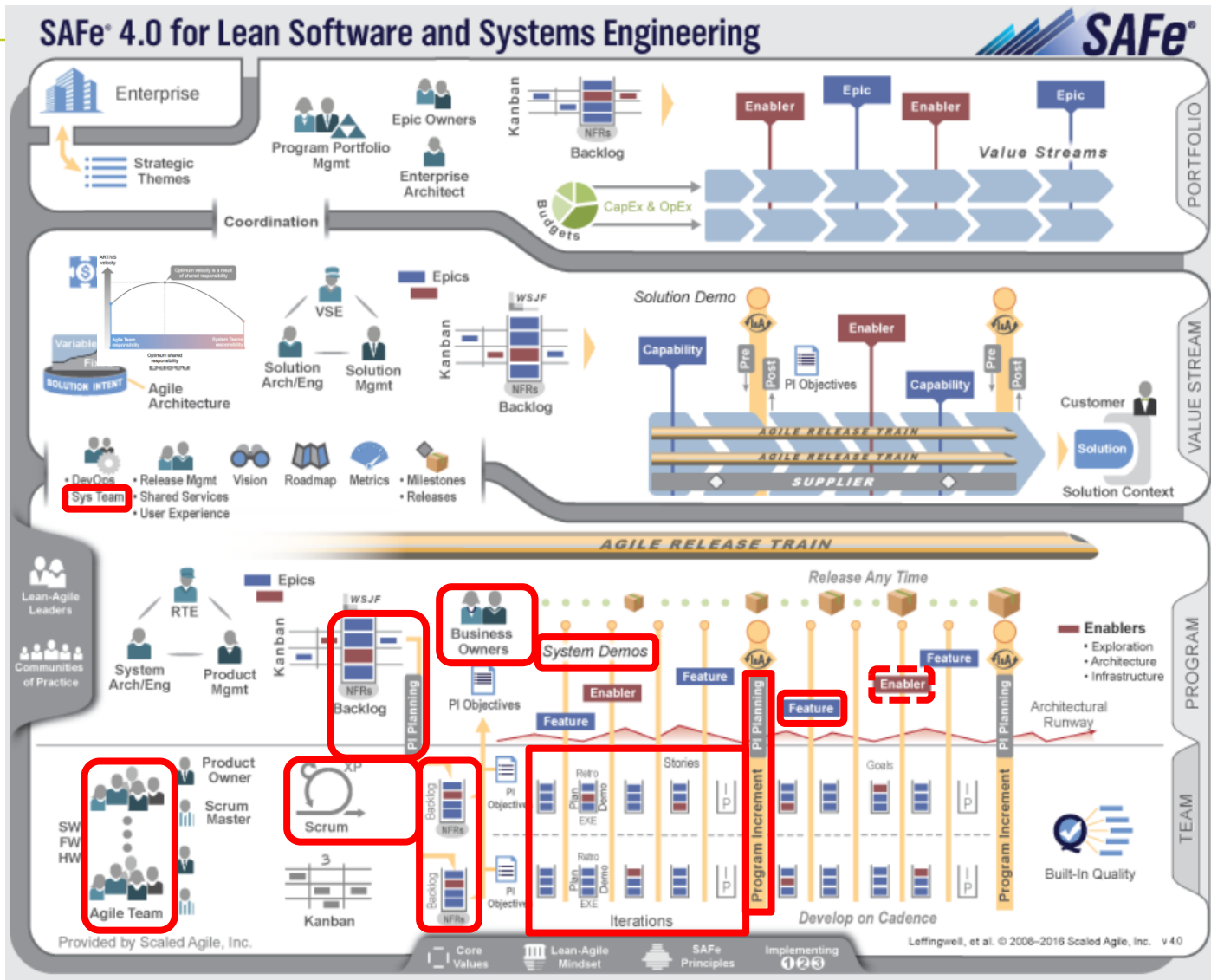


# Our Work

---

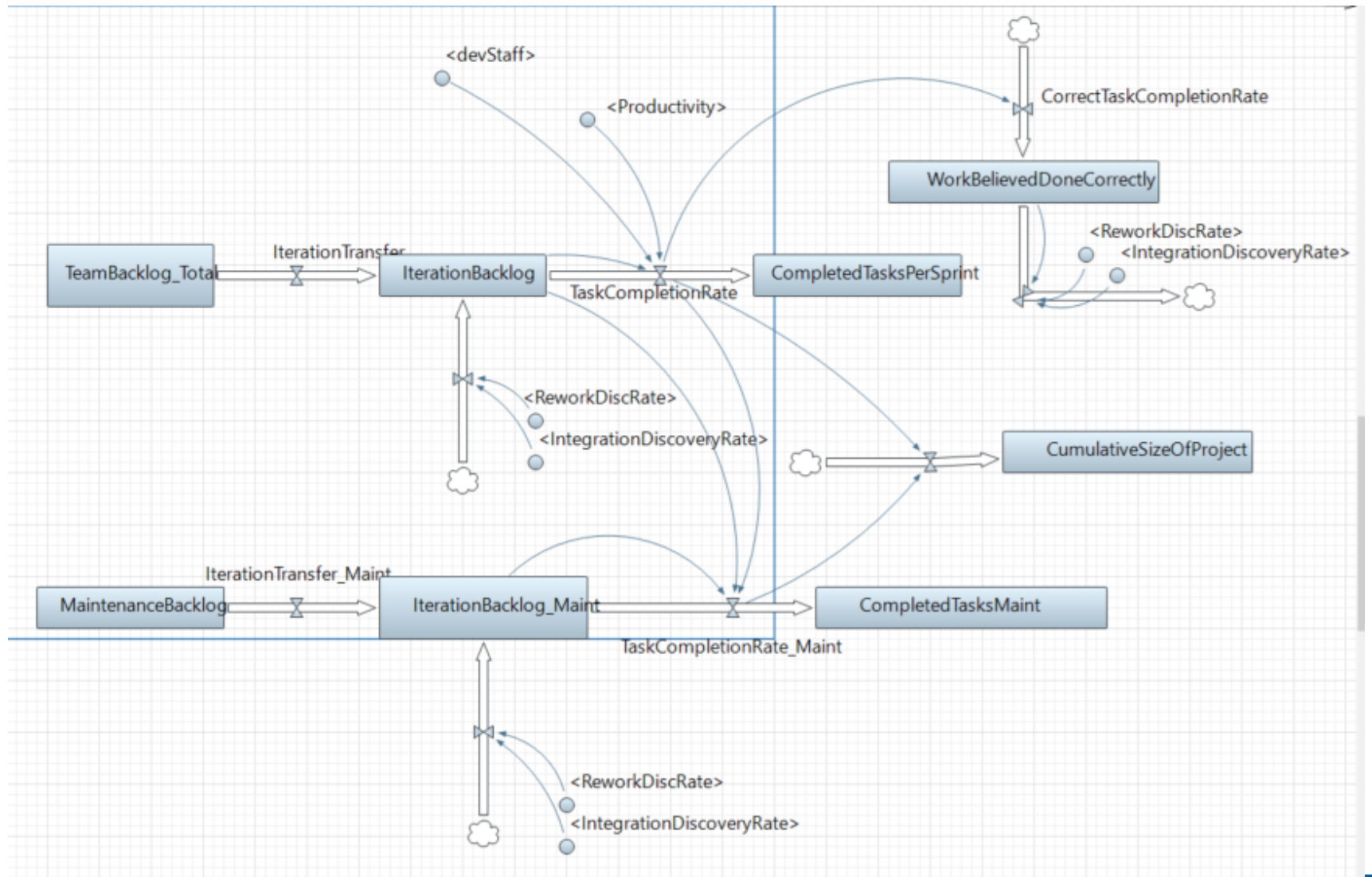
- **Applied to Scaled Agile Framework (SAFe)**
- **Higher level dynamics of team interactions**
- **Extended development cycle to include integration and demos**
- **Distinguish between different types of rework**
  - Defects
  - Integration errors
  - Requirements errors

# SAFe Elements

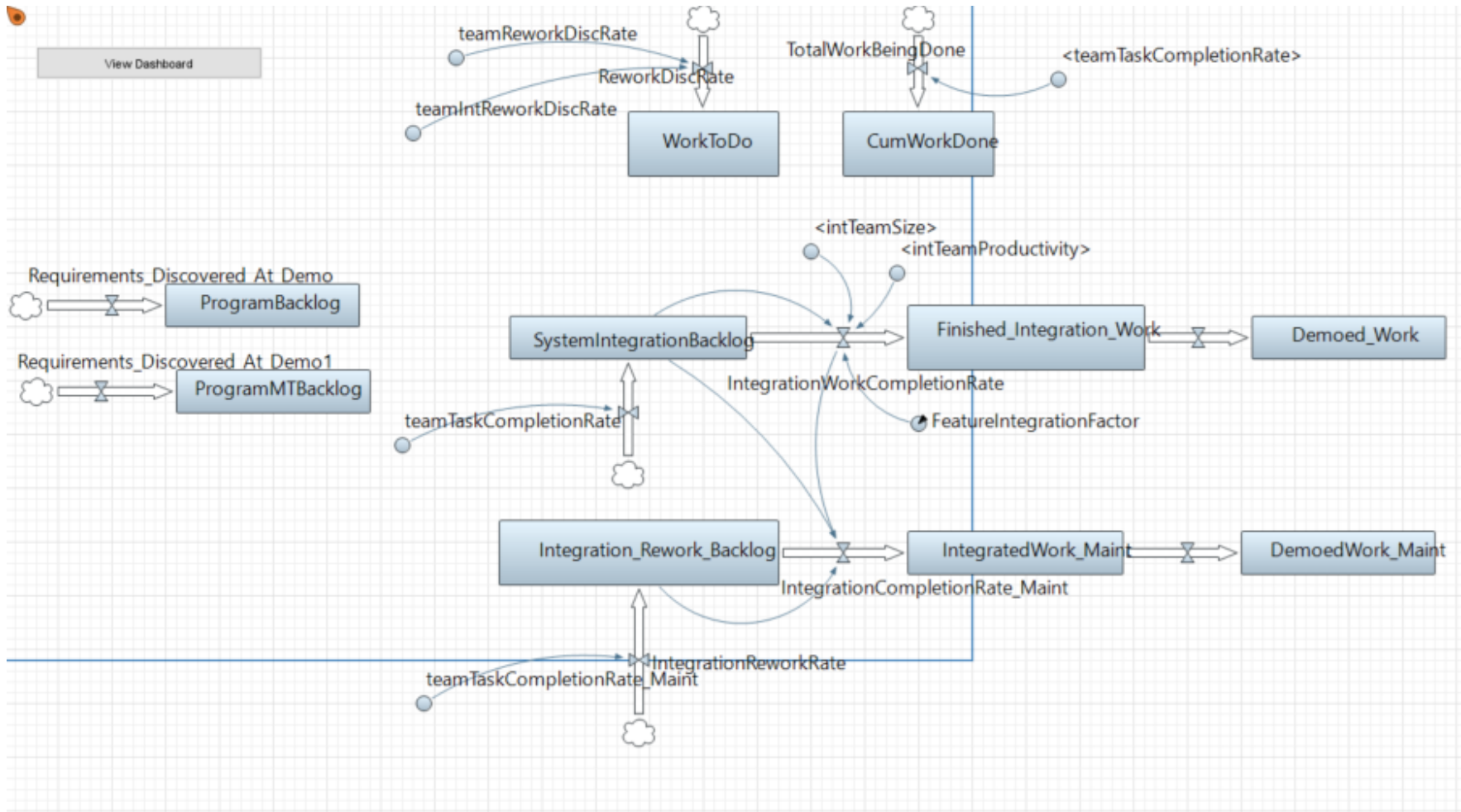


Scaled Agile Framework copyright material used with permission

# Team Work



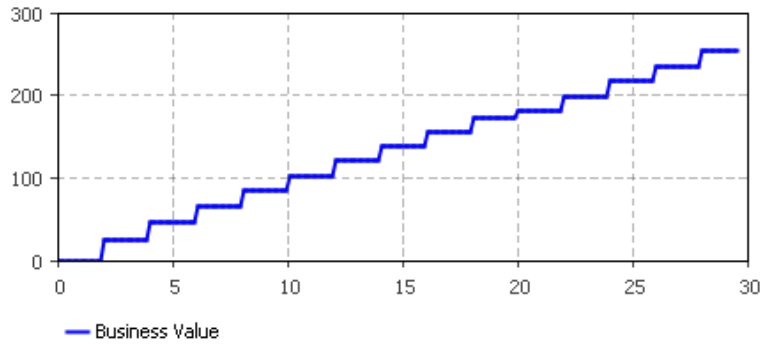
# Program Work



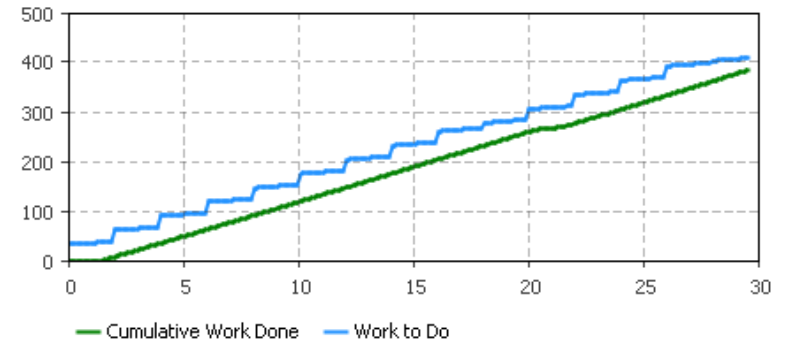
# Output

View Model

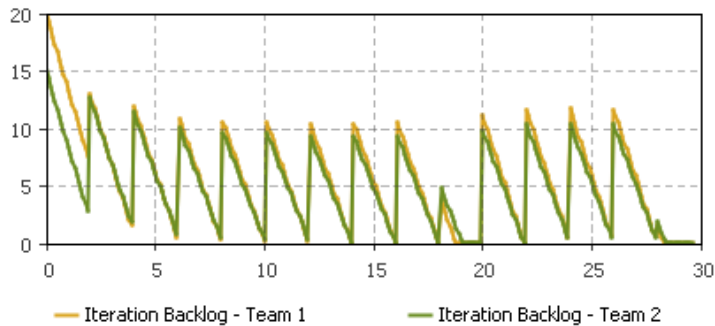
**Total Business Value**



**Cumulative Flow Diagram**



**Iteration Backlog**



Team 1

Team 2

---

# Simulating a Real Project

---

# Case Project Description

## ■ Tailored from SAFe 2.0

- Most team and program elements
- 4 development teams
- 2 weeks per Sprint, 4 Sprints per Program Increment
- No enablers
- No dedicated system team, continuous integration
  - The 4<sup>th</sup> Sprint is used as a development buffer and a time for development teams to do testing and integration work

## ■ Observations

- Large amounts of defects discovered in Sprint 4 leading to delays, cutting into planning sessions, and creating carryover problems for the next Sprint



# Simulation Description

## ■ Without CI (baseline)

- 4 dev teams of 10 each
- No dedicated system team
- 4 \* 2-week Sprints per PI
- Developers do integration during 4<sup>th</sup> Sprint
- 16 PIs simulated

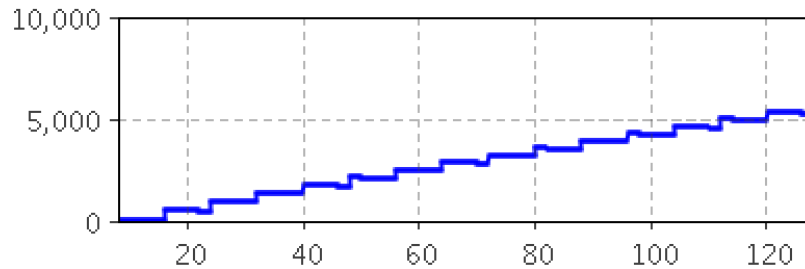
## ■ With CI

- 4 dev teams of 9 each
- Dedicated system team of 4
- 4 \* 2-week Sprints per PI
- All Sprints used for development
- 16 PIs simulated

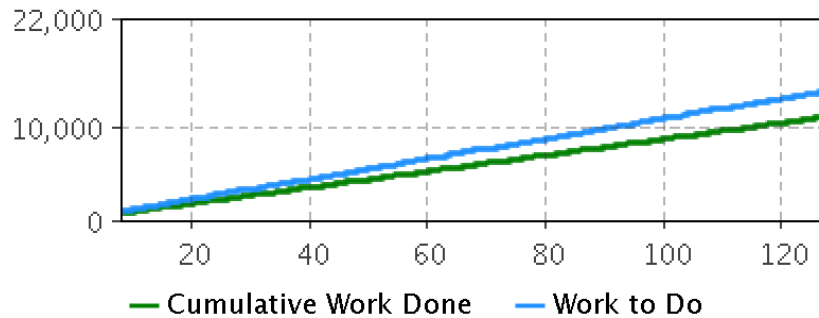
# Results

## ■ Without CI (baseline)

**Total Business Value**

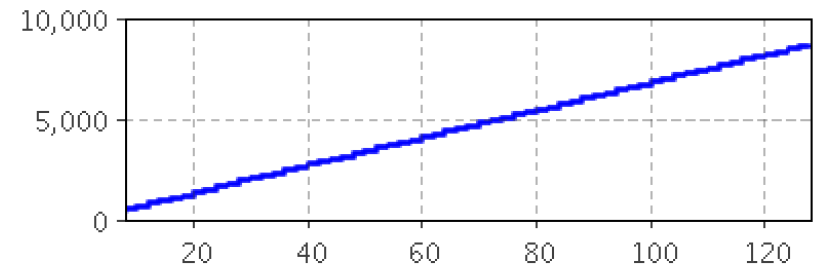


**Cumulative Flow Diagram**

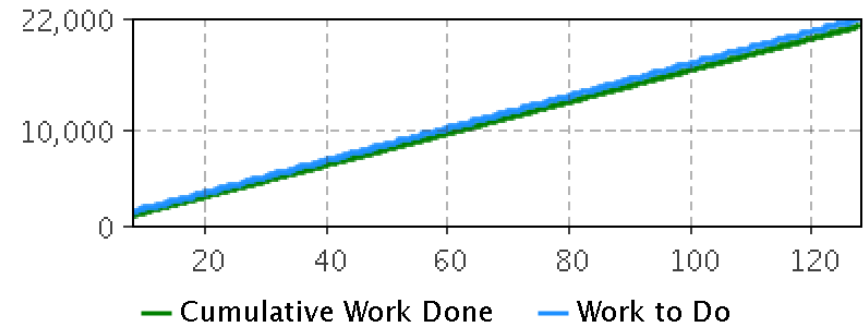


## ■ With CI

**Total Business Value**

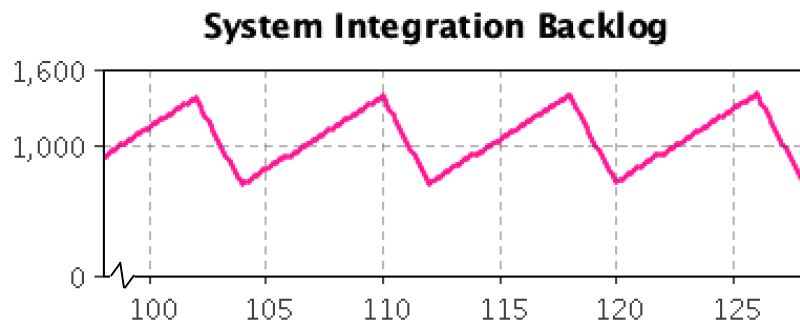


**Cumulative Flow Diagram**

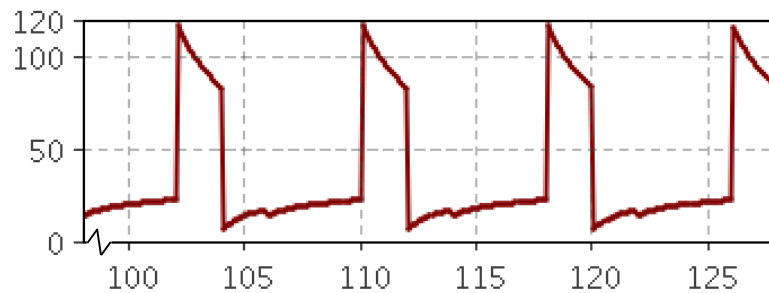


# Results

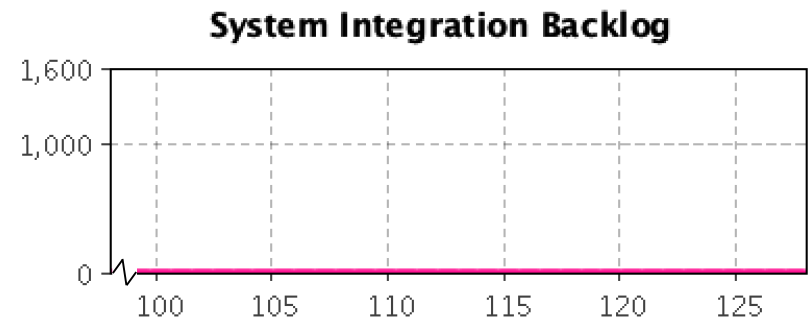
## ■ Without CI (baseline)



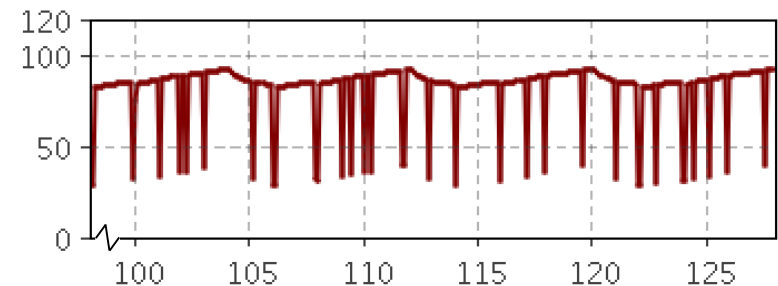
## Rework Discovery Rate



## ■ With CI



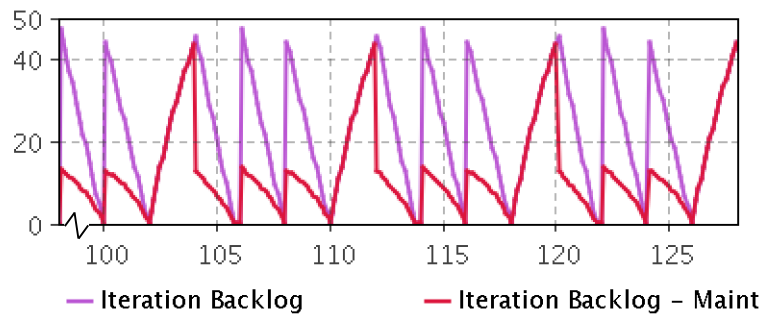
## Rework Discovery Rate



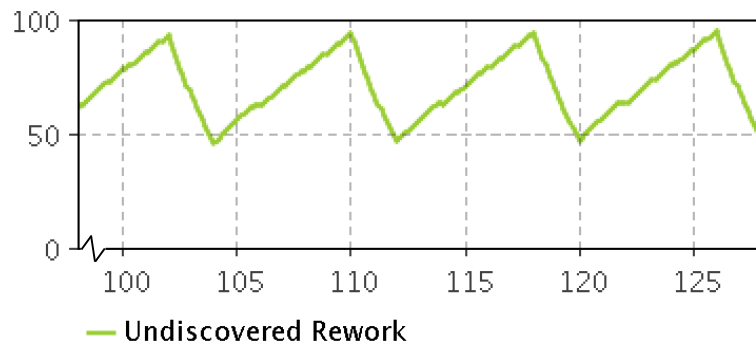
# Results (Team 1)

## ■ Without CI (baseline)

**Backlogs**

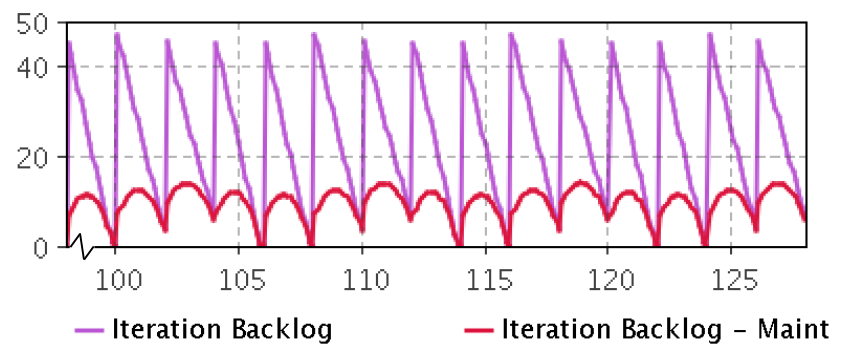


**Undiscovered Rework**

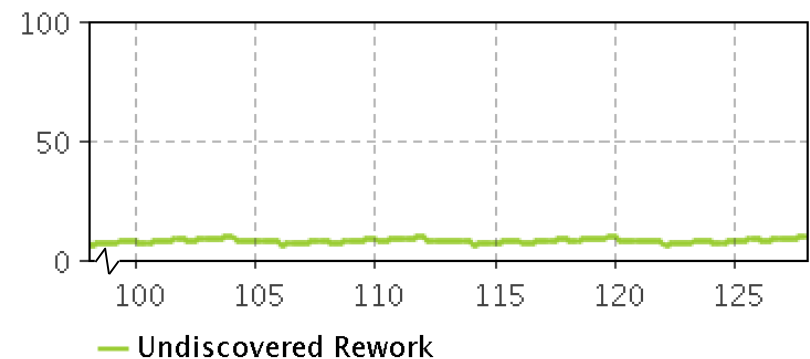


## ■ With CI

**Backlogs**



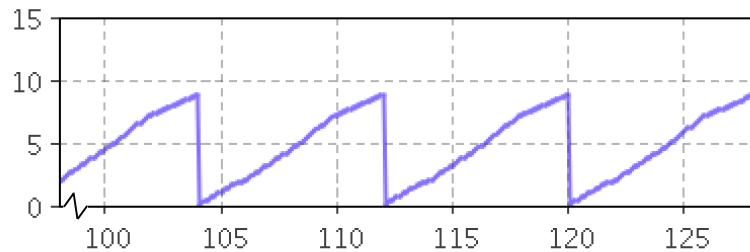
**Undiscovered Rework**



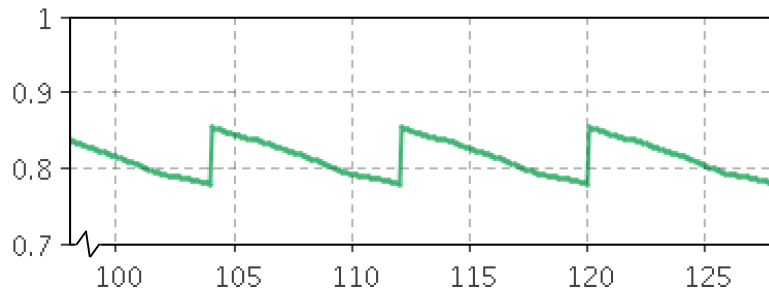
# Results (Team 1)

## ■ Without CI (baseline)

Technical Debt

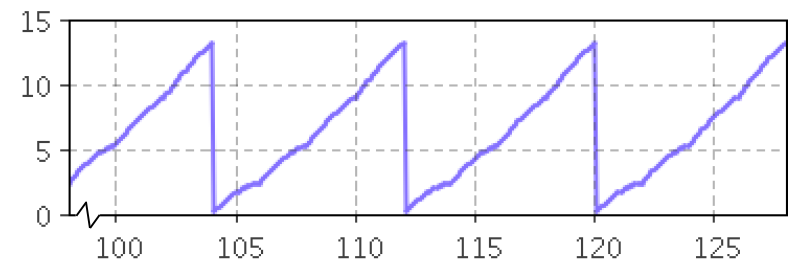


Fraction of Work Correct and Complete

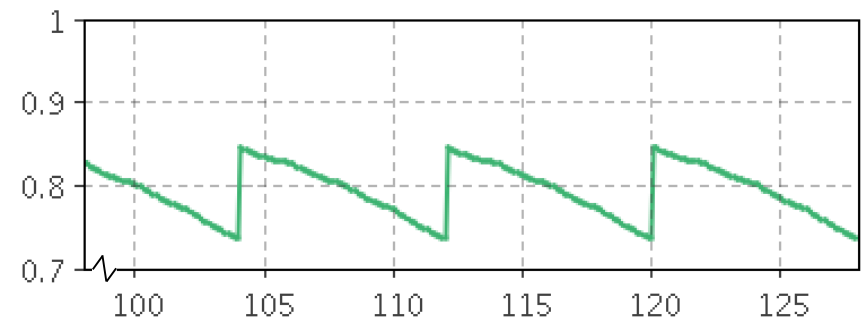


## ■ With CI

Technical Debt



Fraction of Work Correct and Complete



# Results

## ■ Without CI (baseline)

- TBV: 5706
- Average team velocity: 50
- Average undiscovered rework (bugs): 65
- Average FCC: .81

## ■ With CI

- TBV: 8787
- Average team velocity: 49
- Average undiscovered rework (bugs): 8
- Average FCC: .78

**Doing integration continuously rather than waiting until the 4<sup>th</sup> sprint resulted in 54% more valuable work accomplished in the same amount of time with 88% fewer bugs in the code.**

# Limitations

---

- **SAFe or similar programs**
- **Homogenous stocks**
  - Stories and Features
  - Weighted shortest job first (WSJF)
- **Instantaneous meetings**

# Future work

---

- **Improving the model**
  - Generalization
  - Effects of planning sessions
  - Effects of enablers
  - Communication/coordination overhead
- **Verification/Validation**
  - Case studies
  - Sensitivity analysis
- **Management flight simulator**



# Conclusion

---

- **Research builds on work begun at MIT**
- **Identified Agile scaling variables**
- **System dynamics techniques used to model the behavior of complex systems over time**
- **Begun building model for SAFe**
- **Model will provide a decision support tool**

# Agile Dynamics at Scale

## A MITRE Innovation Program

### Research Project

---

# Questions

---



# Thank You!

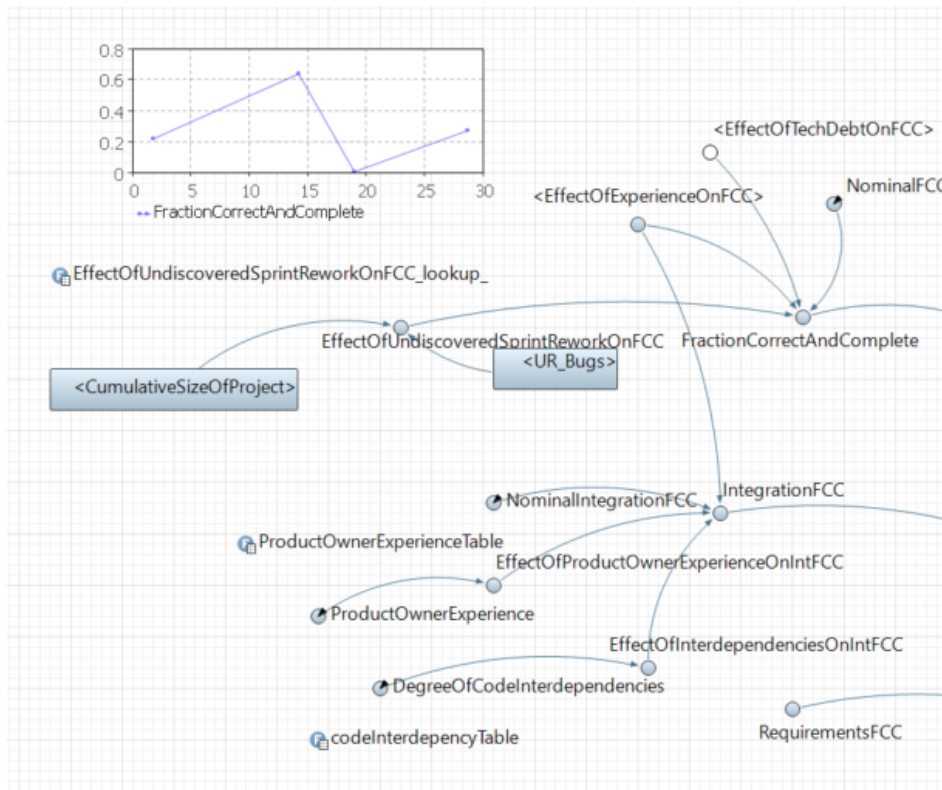
Aleksandra Markina-Khusid [amk@mitre.org](mailto:amk@mitre.org)  
Sean Ricks [stricks@mitre.org](mailto:stricks@mitre.org)

---

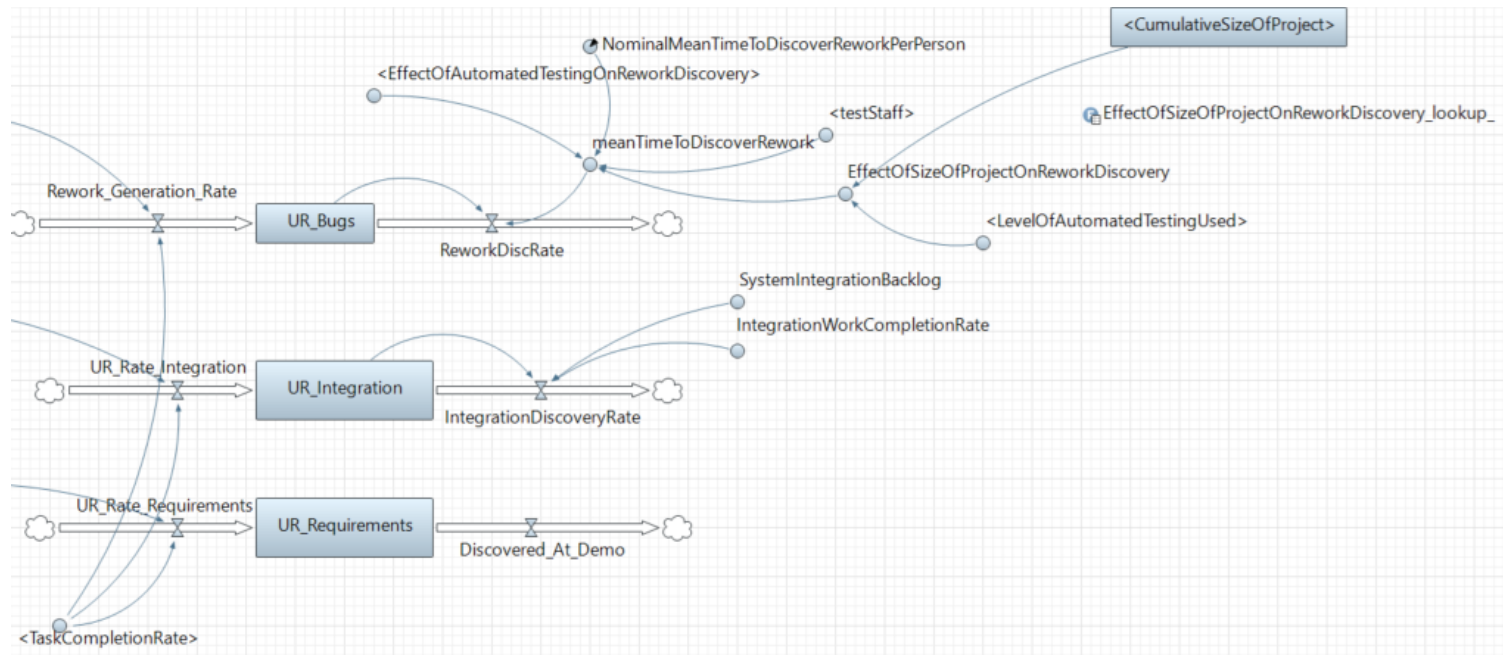
# Backup

---

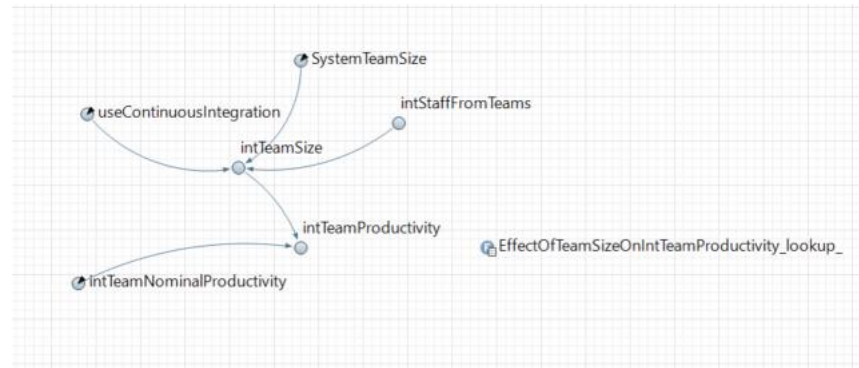
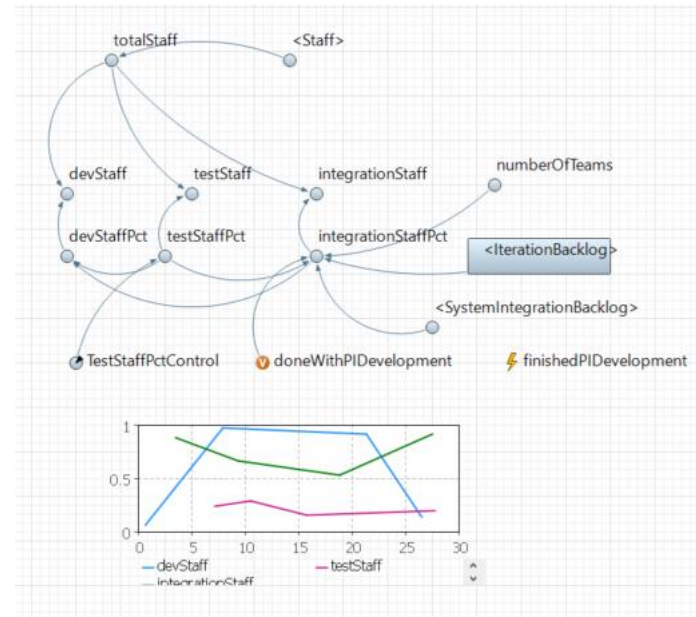
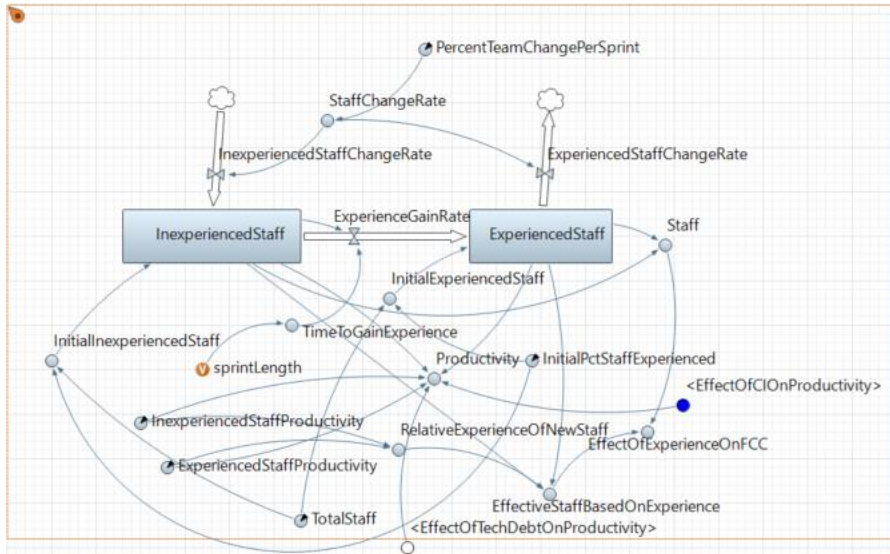
# Fraction Correct and Complete



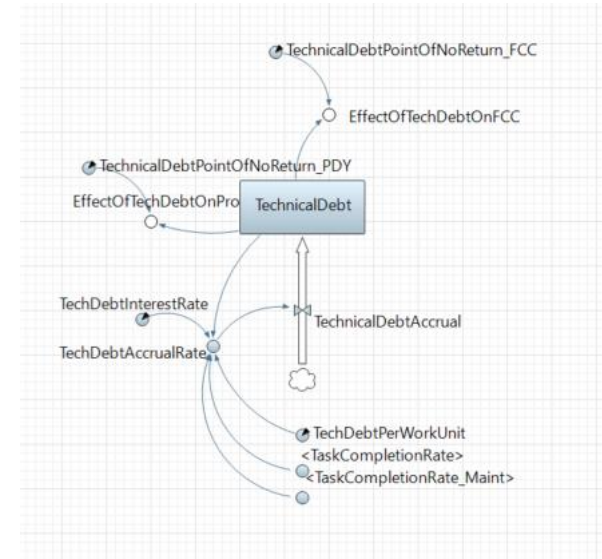
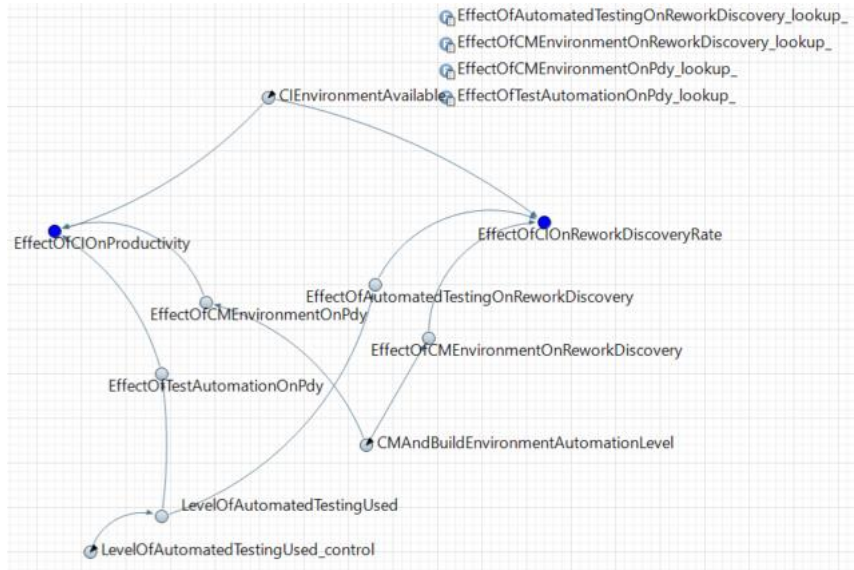
# Rework Creation and Discovery



# Human Resources and Staff Allocation



# Effects of Automation and Tech Debt



# Acronyms

▪ MIT	Massachusetts Institute of Technology
▪ DOD	Department of Defense
▪ APD	Agile Project Dynamics
▪ SAFe	Scaled Agile Framework
▪ EMRAM	Enterprise Modernization Risk Assessment Model
▪ CAMIS	Cadet Administrative Management Information System
▪ AFMC/A4N	Air Force Materiel Command, System Integration Division
▪ DCAPEs	Deliberate and Crisis Action Planning and Execution Segments
▪ SEI	Software Engineering Institute
▪ NDIA	National Defense Industry Association
▪ MDA	Milestone Decision Authority
▪ COR	Contracting Office Representative
▪ PM	Project Manager
▪ FFRDC	Federally Funded Research and Development Center
▪ SME	Subject Matter Expert
▪ SEP	System Engineering Plan
▪ SE	System Engineering
▪ SD	System Dynamics
▪ ALCM	Agile Lifecycle Management
▪ PI	Program Increment
▪ CI	Continuous Integration
▪ TBV	Total Business Value
▪ FCC	Fraction Correct and Complete
▪ WSJF	Weighted Shortest Job First
▪ GOAA	Government Organization Agility Assessment
▪ AiDA	Acquisition in the Digital Age
▪ AF	Air Force