Ricks 19755
Agile Dynamics at Scale
A MITRE Innovation Program
Research Project

NDIA 20th Annual Systems Engineering Conference

Presenting author:
Aleksandra Markina-Khusid 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
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.

▪ 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.

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

Scaled Agile Framework (SAFe) and Agile Scaling Variables

Using System Dynamics to Investigate the Scaled Agile Framework for Lean Software Development
Project Structure

Agile Systems Dynamics

- Literature Survey
- Modeling & Simulation
  - Agile at Scale Parameters & Framework
    - Agile Dynamics
  - Dashboards & Predictive Analytics
    - APD & EMRAM
- Tool Building
- Case Studies
  - CAMIS
- Papers
  - Agile Scaling Variables
    - Using System Dynamics to Investigate the Scaled Agile Framework for Lean Software Development
    - A System Dynamics Model of the Scaled Agile Framework
- Presentations
  - SEI Agile Colloquium
  - NDIA
  - System Dynamics Conference
  - Air Force Information Technology Conference 2017
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*

System Dynamics

Figure 2 - Some Sample Project Dynamics

Prior Work

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

**Prior Work**

### AgileProject

#### Model Parameters

**Continuous Integration Sub-model**
- Configuration Management and Build Environment
- Level of Test Automation
- Number Involved in Setting up the Continuous Integration Environment (Typically a small team of 1 to 3 individuals)

**Customer Involvement Sub-model**
- Maximum Effect of Customer Involvement on Nework Discovery Time
- The Maximum Amount by Which Uncertainty in Customer Requirements Affects Gut Instinct and Complete
- Sensitivity for the Effect of Customer Involvement on Requirements Churn

**Refactoring Sub-model**
- Refactoring Aggressiveness
- Tech Debt Accrued per Unit of Work

**Release Timing Sub-model**
- Number of Software Releases in Agile Project
- Release Planning Duration (in weeks)

**Software and Integration Test Cycle Sub-model**
- Nominal Integration and Test Productivity (The average number of rework tasks/successful)
- Number of Integration and Test Engineers

**Software Development Cycle Sub-model**
- Maximum Work Intensity (Assumes the ability to work 50% extra in overtime)
- Nominal fraction correct and complete (FCC)
- Initial Project Size
- Time for Pressure to Affect FCC (How many weeks of overtime before there is an effect of FCC)
- Time to Discover Rework in Sprint

**Sprint Timing Sub-model**
- Normal Sprint planning duration (in days)
- Sprint duration (in weeks)

**Staffing and Experience Sub-model**
- Initial Experienced Staff
- Initial Inexperienced Staff
- Nominal time to gain experience using waterfall (in weeks)
- Percentage of the team that changes per sprint (fraction)
- Relative experience of new staff compared to old staff

**Team Dynamics Sub-model**
- Nominal effect of pair programming on FCC
- Number of team meetings per week
- Percent of time developers spend pair programming

**Productivity Sub-model**
- Nominal productivity (tasks per person-week)
Prior Work

Dashboard

Status of Controls
- Switch for waterfall: Off
- Allow continuous integration: Off
- Allow customer involvement: Off
- Allow micro-optimization: Off
- Allow pair programming: Off
- Allow refactoring: Off
- Allow integration and test activities: On

Project Finished Day: 73

Approved for Public Release; Distribution Unlimited. Case Number 17-3159 ©2017 The MITRE Corporation. ALL RIGHTS RESERVED.
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

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 4th 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 4th 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)**

- **With CI**

![](image1.png)

![](image2.png)

![](image3.png)

![](image4.png)
Results

- Without CI (baseline)
- With CI

**System Integration Backlog**

<table>
<thead>
<tr>
<th>Without CI</th>
<th>With CI</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="System Integration Backlog" /></td>
<td><img src="image2.png" alt="System Integration Backlog" /></td>
</tr>
</tbody>
</table>

**Rework Discovery Rate**

<table>
<thead>
<tr>
<th>Without CI</th>
<th>With CI</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image3.png" alt="Rework Discovery Rate" /></td>
<td><img src="image4.png" alt="Rework Discovery Rate" /></td>
</tr>
</tbody>
</table>
Results (Team 1)

- **Without CI (baseline)**

- **With CI**

---

**Backlogs**

- **Iteration Backlog**
- **Iteration Backlog – Maint**

---

**Undiscovered Rework**

- **Undiscovered Rework**
Results (Team 1)

- **Without CI (baseline)**

- **With CI**
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 4th 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
  Sean Ricks  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