Development and Quasi-experimental Study of The Scrum Model-Based System Architecture Process (sMBSAP) for Agile Model-Based Software Engineering

Ph.D. Dissertation Defense
Moe Huss | Department of Systems Engineering
July 10th, 2023

Advisor: Dr. Daniel R. Herber
Co-advisor: Dr. John M. Borky
Committee: Dr. Erika Miller
Dr. Paul Mallette
Outline

1. Introduction and Background
2. Implementing the Proposed Integration of Agile and MBSE (Stage 1)
3. Research Methodology (Stage 2)
4. Research Outcome and Results (Stages 1 and 2)
5. Conclusions
Introduction and Background

- Motivation for research
- Software and system development methods (definitions)
- Research gap
- Research purpose (stages)
- System selected for the research
Academic and Professional Background

• **Academic Background**
  – Bachelor of Science, Industrial Engineering from Zagazig University (Egypt), 2005
  – Master of Science, Project Management from the George Washington University, 2016

• **Professional Background**
  – *Oil and Gas Manufacturing and Supply Chain 2000*
    • Project engineer, Project Manager, Program Manager | National Oilwell Varco
  – *Construction Program Management and Supply Chain 2012*
    • Business Process Reengineering Program Manager – 2022 FIFA World Cup
  – *State of California Technology Consulting (Managing Director at iFish Group) 2016*
    • Solution Lead - statewide integrated traffic records system
    • QA Technical Lead – California Statewide Automated Welfare System (CalSAWS)
    • IV&V Technical Lead - System Modernization - CA Department of Public Health
    • Engagement Manager - Cybersecurity Project – CA Department of Finance
What Can the State do to Prevent Future IT Project Delays and Cost Overruns

Oversight Hearing of the Business, Professions and Economic Development Committee’s Subcommittee on California’s Innovation, Technology and Life Sciences Economy

March 26, Upon Adjournment, Room 444

EXAMPLES OF PROBLEM

Many of California’s most prominent tech projects in the past few years have suffered colossal delays and significant cost overruns — more than $2 billion alone for seven big projects since 2011. The state Department of Technology has nine large systems-integration projects in the works right now. The combined value: $3.75 billion.

The state paid out nearly $900 million on three projects before canceling them:

- 21st Century Project — overhaul of the state’s payroll system.
- Department of Motor Vehicles upgrade of the driver’s license and vehicle registration system.
- Judicial Council courts management system

California’s $1 Billion Accounting Software Continues to Plague Taxpayers

Nearly 15 years in, Fi$Cal remains incomplete and over budget while critics warn it could damage California’s pandemic recovery.

OPINION

Walters: Can California shut down failed projects?

Bullet train to nowhere?
Challenges of Software Systems

In collaboration with the University of Oxford, McKinsey studied 5400 large IT projects exceeding $15 million [32].

<table>
<thead>
<tr>
<th>Project type</th>
<th>Average cost overrun</th>
<th>Average schedule overrun</th>
<th>Average benefits shortfall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software</td>
<td>66</td>
<td>33</td>
<td>17</td>
</tr>
<tr>
<td>Nonsoftware</td>
<td>43</td>
<td>3.6</td>
<td>133</td>
</tr>
<tr>
<td>Total</td>
<td>45</td>
<td>7</td>
<td>56</td>
</tr>
</tbody>
</table>

Standish Group CHAOS report 2020 [5]
From DBSE to MBSE

Traditional System Development Methods

**Waterfall:** Software development activities are completed sequentially.

- **Pros:**
  - Requirements

- **Cons:**
  - Implementation

- **Model-Based System Development**

**Document-Based System Development**

Methods that uses documents and multiple tools to capture and communicate the system design throughout the project lifecycle.

SDLC methods, pros and cons [1,18,22,25,38-58,203]
From DBSE to MBSE

Traditional System Development Methods

Waterfall: Software development activities are completed sequentially.

Pros

Cons

Agile System Development Methods

Iterative and incremental approach to software development. It supports iterative development, adaptive planning, and time boxing

Pros

Cons

Document—Based Systems Engineering (DBSE)

Methods that uses documents and multiple tools to capture and communicate the system design throughout the project lifecycle.
Model-Based System Development

**Model-Based Systems Engineering (MBSE)**
Method to create an architecture for a system through structured decomposition into modular and manageable levels of complexity by using object-oriented principles.

**Architecture Frameworks**
- NIST Enterprise Architecture
- TOGAF
- FEAF
- DoDAF
- Zachman

**MBSE Approach**
- IBM Telelogic Harmony-SE
- INCOSE OOSEM
- IBM RUP SE for MDSD
- Vitech MBSE
- JPL State Analysis (SA)
- Dori OPM
- MBSAP

**Agile System Development Methods**
- Agile Modeling
- ASD
- AUP
- Crystal Clear
- DSDM
- FDD
- Lean Software Development

**Traditional System Development Methods**
- Waterfall
- V-Model
- Spiral Model
- RAD
- RUP
- Big Bang Model

Document-Based System Development

**Document-Based Systems Engineering (DBSE)**
Methods that uses documents and multiple tools to capture and communicate the system design throughout the project lifecycle.

**Model—Based Systems Engineering (MBSE)**

**Document-Based System Development**

**SDLC methods, pros and cons [1,18,22,25,38-58,203]
From MBSE to MBSAP

Model-Based Systems Architecture Process (MBSAP)
Method to create an architecture for a system through structured decomposition into modular and manageable levels of complexity by using object-oriented principles.

Model—Based Systems Engineering (MBSE)
Method to create an architecture for a system through structured decomposition into modular and manageable levels of complexity by using object-oriented principles.
From Agile to Scrum

An enhancement of the iterative and incremental approach to delivering object-oriented software.

87% of present-day software industry use Scrum [152].

Document-Based System Development

Methods that uses documents and multiple tools to capture and communicate the system design throughout the project lifecycle.

SDLC methods, pros and cons [1,18,22,25,38-58,203]
Towards Agile MBSE

System Development Lifecycle (SDLC)

Document-Based System Development
- Traditional System Development Methods
  - Waterfall
  - V-Model
  - Spiral Model
  - RUP
  - RAD
  - Big Bang Model

- Agile System Development Methods
  - Agile Modeling
  - ASD
  - AUP
  - Crystal Clear
  - DSDM
  - FDD
  - Lean Software Development
  - SAFe
  - Scrum
  - XP
  - Rapid Prototyping
  - Iterative Model
  - Incremental Model

Model-Based System Development
- Architecture Frameworks
  - NIST Enterprise Architecture
  - TOGAF
  - FEAF
  - DoDAF
  - Zachman

- Agile MBSE Approach
  - May leverage an architecture framework (for modeling), and combined MBSE and an Agile methods (for development process)

MBSE Approach
- IBM Telelogic Harmony-SE
- INCOSE OOSEM
- IBM RUP SE for MDSD
- Vitech MBSE
- JPL State Analysis (SA)
- Dori OPM
- MBSAP
Limited publicly available measured evidence of the potential benefits of adopting and implementing MBSE* [192].

No experiments that make a side-by-side comparison between Agile MBSE and Agile.

No practical guide found for implementing Agile MBSE for Software Systems [22-25], [204], [207].
Research Purpose (Stage 1)

System Development Lifecycle (SDLC)

Document-Based System Development

Traditional System Development Methods
- Waterfall
- V-Model
- Spiral Model
- RAD
- Big Bang Model

Agile System Development Methods
- Agile Modeling
- ASD
- AUP
- Crystal Clear
- DSDM
- FDD
- Lean Software Development

Model-Based System Development

Architecture Frameworks
- NIST Enterprise Architecture
- TOGAF
- FEA\F
- DoDAF
- Zachman

Agile MBSE Approach
May leverage an architecture framework (for modeling), and combined MBSE and an Agile methods (for development process)

MBSE Approach
- IBM Telelogic Harmony-SE
- INCOSE OOSEM
- IBM RUP SE for MDSD
- Vitech MBSE
- JPL State Analysis (SA)
- Dori OPM
- MBSAP

Scrum MBSAP (sMBSAP)
Research Purpose (Stage 2)

Comparing sMBSAP and Scrum, are there measurable benefits to software development performance when using one approach over the other?

First Hypothesis

Hypothesis H₀: sMBSAP has equivalent or better Reliability of Estimation [60-63] relative to Scrum, as assessed by Commitment Reliability (CR) [74-77], that is, the ratio of work delivered vs. work committed during a sprint, measured in Story Points (SP) [99].

Hypothesis H₁: sMBSAP has inferior Reliability of Estimation [60-63] relative to Scrum, as assessed by Commitment Reliability (CR) [74-77], that is, the ratio of work delivered vs. work committed during a sprint, measured in Story Points (SP) [99].

Second Hypothesis

Hypothesis H₀: sMBSAP has equivalent or better Productivity [98-99] relative to Scrum, as assessed by (a) Sprint Velocity (SV) [103-105], that is, the amount of work delivered during a sprint, measured in Story Points (SP) [99], (b) Velocity Fluctuation (VF) [109] that is the deviation from the average Sprint Velocity, and (c) Count of Lines of Code (CLOC) per hour [99].

Hypothesis H₁: sMBSAP has inferior Productivity [98-99] relative to Scrum, as assessed by (a) Sprint Velocity (SV) [103-105], that is, the amount of work delivered during a sprint, measured in Story Points (SP) [99], (b) Velocity fluctuation (VF) [109] that is the deviation from the average Velocity, and (c) Lines of Code (LOC) per hour [99].

Third Hypothesis

Hypothesis H₀: sMBSAP has equivalent or lower Defect Rate [110-112] relative to Scrum, as assessed by (a) Defect Density (DD) [116-118], that is, the number of defects found within Sprint Backlog Items (SBIs) [64, 120] or Thousands (Kilo) of Lines of Code (KLOC) [118], and (b) Defect Leakage (DL) [122-125], that is, the ratio of defects found during testing vs. defects found during the next phase, measured in defects count.

Hypothesis H₁: sMBSAP has higher Defect Rate relative to Scrum, as assessed by (a) Defect Density (DD), that is, the number of defects found within Sprint Backlog Items (SBIs) or Thousands (Kilo) of Lines of Code (KLOC), and (b) Defect Leakage (DL), that is, the ratio of defects found during testing vs. defects found during the next phase, measured in defects count.
System Selected for the Experiment

The development of a health technology system that aims at providing diet and lifestyle recommendations to users was used to investigate the research question.

Scope of the System Case Study

- Registration
- Health Assessment
- Health Reporting
- Scientific Evidence
- Grocery Recommendations
- Order Groceries
- Track Grocery Consumption
- Monitor Health Progress

Scope of architecture limited for research:

- Single reference architecture
- Primarily software category
- Four modules and one subsystem
- Operational to physical level of abstraction
- The physical product is a prototype

The four architecture taxonomic dimensions [1]
Product Vision

We are what we eat,

https://www.youtube.com/watch?v=mvDRgxgbYkM
Implementing the Proposed Integration of Agile and MBSE (sMBSAP) | Stage 1

• Overview of sMBSAP
• How does sMBSAP work?
Agile (Scrum) + MBSE (MBSAP) = Agile MBSE (sMBSAP)

System Development Lifecycle (SDLC))

Document-Based System Development

Traditional System Development Methods
- Waterfall
- V-Model
- Spiral Model
- RUP
- RAD
- Big Bang Model

Agile System Development Methods
- Agile Modeling
- ASD
- AUP
- Crystal Clear
- DSDM
- FDD
- Lean Software Development

SAFe
Scrum
XP
Rapid Prototyping
iterative Model
Incremental Model

Model-Based System Development

Architecture Frameworks
- NIST Enterprise Architecture
- TOGAF
- FEA
- DoDAF
- Zachman

Agile MBSE Approach
May leverage an architecture framework (for modeling), and combined MBSE and an Agile methods (for development process)

MBSE Approach
- IBM Telelogic Harmony-SE
- INCOSE OOSEM
- IBM RUP SE for MDSD
- Vitech MBSE
- JPL State Analysis (SA)
- Dori OPM
- MBSAP
Agile (Scrum) + MBSE (MBSAP) = Agile MBSE (sMBSAP)

SDLC methods, pros and cons [1,18,22,25,38-58,203]
Agile (Scrum) + MBSE (MBSAP) = Agile MBSE (sMBSAP)

sMBSAP Meetings

- Sprint Retro

sMBSAP Artifacts

- Physical Viewpoint
- Increment / Demo
- Product/Sprint Backlog
- Operational Viewpoint
- Daily Standups

sMBSAP Processes

- Initiate
- Plan and Architect
- Implement
- Review and Retrospect
- Release

sMBSAP Roles

- Product Owner
- Scrum Master
- System Architect
- Development Team

sMBSAP Artifacts

- Logical/Functional Viewpoint
- Increment / Demo
- Operational Viewpoint
- Daily Standups

MBSAP

- Capabilities
- Integration and Test
- Increment / Demo
- Logical/Functional Viewpoint
- Physical Viewpoint

Scrum Meetings

- Sprint Retrospective
- Sprint Planning
- Sprint Review
- Sprint Demo
- Sprints

Scrum Roles

- Product Owner
- Team
- Scrum Master

Scrum Processes

- Initiating
- Planning
- Implementing
- Reviewing
- Releasing

SDLC methods, pros and cons [1,18,22,25,38-58,203]
Agile (Scrum) + MBSE (MBSAP) = Agile MBSE (sMBSAP)

- **sMBSAP Meetings**
  - Sprint Retro
- **sMBSAP Artifacts**
  - Physical Viewpoint
  - Increment / Demo
  - Product/Sprint Backlog
  - Sprint / Architecture Planning
- **sMBSAP Roles**
  - System Architect
  - Product Owner
  - Scrum Master
- **Operational Viewpoint**
- **Daily Standups**

**sMBSAP Processes**
- Initiate
- Plan and Architect
- Implement
- Review and Retrospect
- Release

Engaging the team in customizing the tool

Modify the meetings to include architecture planning and review

Using UML-based and non-UML-based models to describe the system

Unified Modeling Language (UML) is a general-purpose modeling language that is intended to provide a standard way to visualize the design of a system [182].

- Modified to integrate Scrum and MBSAP
- Inherited from MBSAP
- Inherited from Scrum
- Common in both Scrum and MBSAP

**Inputs**
- Prioritized Product Backlog
- System Architecture
- Overview and Summary
- Release Plan
- Core Project Team

**Outputs**
- Personas
- Prioritized Product Backlog
- System Architecture
- Overview and Summary
- Release Plan
- Core Project Team

**Outputs**
- Acceptance criteria of the Product Backlog
- Sprint Backlog
- System Architecture
- Overview and Summary
- Release Plan
- Core Project Team

**Inputs**
- Acceptance criteria of the Product Backlog
- Sprint Backlog
- System Architecture
- Overview and Summary
- Release Plan
- Core Project Team

**Outputs**
- Working Deliverables
- Lessons Learned for Future Implementation

**Inputs**
- Prioritized Product Backlog
- Updated System Architecture
- Sprint Deliverables
- Burndown Charts

**Outputs**
- Accepted System Architecture
- Accepted Deliverables
- List of Actionable Improvements

**Outputs**
- Working Deliverables
- Lessons Learned for Future Implementation

**Inputs**
- Acceptance criteria of the Product Backlog
- Sprint Backlog
- System Architecture

**Inputs**
- Acceptance criteria of the Product Backlog
- Sprint Backlog
- System Architecture

**Unified Modeling Language (UML)** is a general-purpose modeling language that is intended to provide a standard way to visualize the design of a system [182].
Progression of Formal UML Models for the Health Tech System

Conceptual Data Model

- Translate the Project Scope Statement, and Product Backlog into an architectural mode.
- A high-level view of the architecture.
- Define the system’s boundary and context.
- Create top-level partitioning (Domains), primary Behaviors (Use Cases), and primary data content.
Progression of Formal UML Models for the Health Tech System

**Logical Data Model**

- Define the details of system elements, functions, and data.
- Progressive elaboration on the perspectives of the OV.
- Functional service specifications are developed and allocated to logical components and interfaces.
- Functional definition of the technology- and product-agnostic system.
Progression of Formal UML Models for the Health Tech System

Conceptual Data Model

- Basis for the actual implementation of the full system or an increment of the system.
- PV defines how the system will be realized.
- Focus on products and standards.

Logical Data Model

Physical Data Model
Progression of Formal UML Models for the Health Tech System

Logical Data Model

Conceptual Data Model

Perspectives (Structure, Data, Behavior, Capabilities)

Viewpoints (Operational, Logical, and Physical)
Integrating Informal Artifacts of the Health Tech System into the Model

Visual Use Case diagrams which also includes user stories and requirements

Text-based sprint log (user stories and requirements)

AWS Architecture Model

* The mostly complete model is shown in Appendix A of the dissertation
From sMBSAP artifacts to products

Architecture Framework

<table>
<thead>
<tr>
<th>Perspective</th>
<th>Structure</th>
<th>Data</th>
<th>Behavior</th>
<th>Capabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conceptual</td>
<td>Class Diagram</td>
<td>Conceptual Data Model</td>
<td>Activity Diagram</td>
<td>User Stories</td>
</tr>
<tr>
<td></td>
<td>Product Breakdown</td>
<td>Structure, Element Hierarchy, Domain Modeling, Structural Decomposition</td>
<td>Functional Flow</td>
<td>Functions, Features, Tasks, Enhancements</td>
</tr>
<tr>
<td>Logical</td>
<td>Logical Data Model</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Physical</td>
<td>Physical Data Model</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Object Diagram</td>
<td>Structural Decomposition</td>
<td>Sequence Diagram</td>
<td>Sprint Backlog</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Objects interaction</td>
<td>Prioritized user stories</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Use Case Diagram</td>
<td>Deprioritized user stories</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Objects interaction</td>
<td></td>
</tr>
</tbody>
</table>

AWS Deployment

+123k Lines of Code Written
From sMBSAP artifacts to products
Product Demo

https://www.youtube.com/watch?v=L0X3ha2TeDY
Research Methodology for an Experiment to Compare sMBSAP and Agile | Stage 2

• Experiment Overview
• Experiment Design
• Techniques and Procedures
• Quality of Research Design
**Experiment Overview**

**Setting:** Software Development Process for a Health Tech System

**System Engineering Approach (x) (independent variables)**
- sMBSAP (A)
- Traditional Scrum (B)

**Groups**
- 10 Scrum-Driven Sprints (control group)
- 10 sMBSAP-Driven Sprints (treatment group)

**System Development Performance (y) (dependent variables)**
- Reliability of Estimation
- Defect Rate
- Productivity

**Experiment**
Comparing the effect of using system engineering approach A and B (x) on system development performance (y)

**Confounding Variables**
- Development Team Experience
- System Architect Experience
- Consistency of Input Artifacts
- Complexity of Work

**Metrics**
- Commitment Reliability (CR)
- Defect Density using PBI
- Defect Density using KLOC
- Defect Leakage (DL)
- Sprint Velocity
- Velocity Fluctuation (VF)
- CLOC per Hour
Experiment Design

After the execution of sprints and collection of data...
(a.) 20 sprints were analyzed for Reliability of Estimation and Defect Rate
(b.) 19 sprints were analyzed for Productivity (a one-week sprint was excluded to avoid skewing the data)

Product Backlog
User Stories (m=260)

Sprints (n=20)

Group 1
Sprints (n=10)
Scrum
Sprints (n=10)

Group 2
Sprints (n=10)
sMBSAP
Sprints (n=10)

Sprints analyzed for effect on dependent variables

Reliability of Estimation
Scrum (n=10) sMBSAP (n=10)

Defect Rate
Scrum (n=10) sMBSAP (n=10)

Productivity
Scrum (n=10) sMBSAP (n=9)

① All user stories (m=260) are identified and added to a product backlog

② Product backlog was broken down and scheduled...
(a.) user stories (m=260) were grouped into sprints (n=20)
(b.) sprints (n=20) were scheduled to deliver the product features

③ Two groups of sprints were created...
(a.) constrained by the schedule to avoid impacting productivity
(b.) to contain equal numbers of sprints
(c.) to avoid the Hawthorne effect by using intact groups
(d.) then one group was randomly assigned to be the treatment group (sMBSAP) and the other as the control group (Scrum).
Techniques and Procedures (1 of 5)

<table>
<thead>
<tr>
<th>Perspective</th>
<th>Structure</th>
<th>Data</th>
<th>Behavior</th>
<th>Capabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operational</td>
<td>Classifications: Hierarchical Levels</td>
<td>Classifications: Hierarchical Levels</td>
<td>Use Case (UC) Diagram: Hierarchical Levels, User Story Types</td>
<td>Classifications: Hierarchical Levels, User Story Types</td>
</tr>
<tr>
<td></td>
<td>Product Breakdown: Conceptual Data Model (CDM): Primary Data</td>
<td></td>
<td>Activity Diagram (AD): with multiple Domains and Actors</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Structure, Structural Decompositions</td>
<td>Logical Data Model (LDM): Modeling Foundation Classes, Attributes and Operations</td>
<td>Sequence Diagram (SD): Modeling non-stateful behavior of a group of classes</td>
<td></td>
</tr>
<tr>
<td>Functional</td>
<td>Class Diagrams: Domain Diagram with Specifications</td>
<td>Physical Data Model (PDM): Database Structure</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Physical</td>
<td>Decomposition of Domains to the level of Classes</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Organizational overview of an information model for the UML-based sMBSAP [2]
Techniques and Procedures (2 of 5)

Metrics for Reliability of Estimation

\[ CR = \frac{\text{Completed SP}}{\text{Planned SP}} \]

Average \( CR = \frac{\sum CR}{\text{Number of Sprints}} \)

Metrics for Productivity

\[ SV = \sum \text{Completed SP During a Sprint} \]

Average Velocity = \[ \frac{\sum \text{Sprint Velocity}}{\text{Number of Sprints}} \]

\[ VF = \frac{\text{Average Velocity} - \text{Sprint Velocity}}{\text{Average Velocity}} \]

\[ \text{CLOC per hour} = \frac{\text{CLOC}}{\text{Development Duration (hrs)}} \]

Identify the Metrics for the Performance of Software Development

② Identify the Metrics for the Performance of Software Development

Metrics for Defect Rate

\[ DD = \frac{\text{Defect Counts (pre-delivery and post-delivery)}}{\text{Size (measured in PBI)}} \]

Average \( DD = \frac{\sum DD}{\text{Number of Sprints}} \)

\[ DD = \frac{\text{Defect Counts (pre-delivery and post-delivery)}}{\text{Size (measured in KLOC)}} \]

\[ DL = \frac{\text{Defect Count (post-delivery)}}{\text{Defect Count (pre-delivery)}} \]

Average \( DL = \frac{\sum DL}{\text{Number of Sprints}} \)

Metrics for Reliability of Estimation [74-77, 99]; Metrics for Productivity [99, 109, 103-105], and Defect Rate [64, 116-118, 120, 122, 125]
Techniques and Procedures (3 of 5)

3. Execute Scrum and sMBSAP Drive-Sprints

Scrum

Architecture/Artifacts framework

Develop artifacts and implementing user stories

Write code
**Techniques and Procedures (4 of 5)**

- **Plan experiment**
- **Identify the metrics for the software development performance objectives**
- **Analyze data / Compare results**
- **Execute Scrum and sMBSAP-driven sprints**
- **Collect software development performance data**

---

**Table B.1: Monitoring CBR for Scrum-driven sprints.**

<table>
<thead>
<tr>
<th>Sprint #</th>
<th>Start Date</th>
<th>End Date</th>
<th>Planned SP</th>
<th>Completed SP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sprint 1</td>
<td>1/1/2020</td>
<td>2/28/2020</td>
<td>34</td>
<td>27</td>
</tr>
<tr>
<td>Sprint 2</td>
<td>3/5/2020</td>
<td>4/22/2020</td>
<td>53</td>
<td>28</td>
</tr>
<tr>
<td>Sprint 3</td>
<td>4/12/2020</td>
<td>5/9/2020</td>
<td>71</td>
<td>26</td>
</tr>
<tr>
<td>Sprint 4</td>
<td>5/12/2020</td>
<td>6/3/2020</td>
<td>35</td>
<td>27</td>
</tr>
<tr>
<td>Sprint 5</td>
<td>6/1/2021</td>
<td>7/4/2021</td>
<td>52</td>
<td>28</td>
</tr>
<tr>
<td>Sprint 6</td>
<td>7/12/2021</td>
<td>8/5/2021</td>
<td>54</td>
<td>26</td>
</tr>
</tbody>
</table>

**Table B.2: Monitoring CBR for sMBSAP-driven sprints.**

<table>
<thead>
<tr>
<th>Sprint #</th>
<th>Start Date</th>
<th>End Date</th>
<th>Planned SP</th>
<th>Completed SP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sprint 1</td>
<td>9/1/2020</td>
<td>9/25/2020</td>
<td>39</td>
<td>27</td>
</tr>
<tr>
<td>Sprint 2</td>
<td>9/26/2020</td>
<td>10/11/2020</td>
<td>36</td>
<td>24</td>
</tr>
<tr>
<td>Sprint 3</td>
<td>10/12/2020</td>
<td>11/15/2020</td>
<td>34</td>
<td>22</td>
</tr>
<tr>
<td>Sprint 4</td>
<td>11/15/2020</td>
<td>12/18/2020</td>
<td>34</td>
<td>22</td>
</tr>
</tbody>
</table>

**Comparing sMBSAP and Agile methods [3]**

* Sprint logs and burnup charts are available in Appendices B and C

---

**Scrum Data**

- **Reliability of Estimation**
- **Productivity**
- **Defect Rate**

**sMBSAP Data**
Techniques and Procedures (5 of 5)

(1) Plan experiment
(2) Identify metrics for software development performance objectives
(3) Execute Scrum and sMBSAP-driven sprints
(4) Collect software development performance data
(5) Analyze data / Compare results

Reliability of Estimation

Productivity

Defect Leakage

Comparing sMBSAP and Agile methods [3]

* Additional raw data and statistical analysis are available in Appendix D
Quality of Research Design

Minimizing Threat to Reliability
(1) Participant error.
(2) Participant bias.
(3) Researcher error.
(4) Researcher bias.

Minimizing Threat to Validity (internal validity)
(1) Selection bias.
(2) Testing effect.
(3) History effect.
(4) Instrumentation effect.
(5) Mortality effect.

Minimizing Threat to Validity (external validity)
(1) Changes of causal relationship due to variations of the implementation within the same approach.
(2) Interaction of causal relationship with settings.
(3) Interaction of causal relationship with outcomes.
(4) The reactive or interactive effect of testing.

Minimizing Threat to Statistical Conclusion Validity
(1) Random assignment.
(2) Independence.
(3) Level of measurement.
(4) Normality and outliers.
(5) Homogeneity of variance.
Research Outcome and Results

Stage 1: sMBSAP Implementation
Stage 2: Experiment Results
sMBSAP Implementation

<table>
<thead>
<tr>
<th>Focus</th>
<th>Scrum</th>
<th>MBSAP</th>
<th>sMBSAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collaboration, Iterative and Incremental Development, Continuous Improvement, Customer-Centricity, Prioritizing Delivering Value to Customer</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Approach | Rapid iteration | Rapid or linear | Rapid or linear |

| Project's Scale | Small and medium | Medium or large | Small, medium or large |

| Application | Software Development and Delivering Working Software | Systems Engineering and the Creation of High-Level System Models | Application agnostic |

| Structure, Data, Behavior and Requirements | Informal, when Necessary, Document-Based | Formal, Necessary, Mode-Based | Formal and Informal, when Necessary, Mode-Based |

| Roles | Product Owner, Scrum Team, and Scrum Master | Program/Project Manager, System Architect, Project Team | Product Owner, Scrum Team, Scrum Master, and System Architect |

| Ceremonies | Daily Standups, Sprint Retro, Sprint Planning, Sprint Review | Architecture kickoff Workshop, Formal and Informal Program Reviews and Coordination Meetings | Daily Standups, Sprint Retro, Sprint/Architecture Planning, Spring/Architecture Review |


Build on and leverage the best capabilities of both methods.

Find a common ground of communication and collaboration between stakeholders with different backgrounds.
Hypothesis 1: Reliability of Estimation

Commitment Reliability (CR)

- **Scrum**
  - Highest CR = 0.86
  - Lowest CR = 0.71
  - M = 0.81 (SD = 0.046)

- **sMBSAP**
  - Highest CR = 1.0
  - Lowest CR = 0.87
  - M = 0.93 (SD = 0.032)

sMBSAP-driven sprints were associated with a statistically significant larger mean CR than the Scrum-driven sprints.
Hypothesis 2: Productivity

\[ CLOC \text{ per hour}(\text{Scrum}) = 58.03 \]
\[ CLOC \text{ per hour}(s\text{MBSAP}) = 71.1 \]

sMBSAP-driven sprints were associated with a statistically significant larger mean SV than the Scrum-driven sprints.

The results from this study suggest no noticeable difference in the VF between the Scrum and sMBSAP-driven sprints.
Hypothesis 3: Defect Rate

\[ DD \text{ in KLOC (Scrum)} = 4.57 \]
\[ DD \text{ in KLOC (sMBSAP)} = 2.24 \]

sMBSAP-driven sprints were associated with a statistically significant smaller mean \( DD \) (using PBIs) than the Scrum-driven sprints.

sMBSAP-driven sprints were associated with a numerically smaller \( DL \) than the Scrum-driven sprints.

\[ \text{Defect Leakage} \]

\[ \text{Difference between means} \]

\[ \text{95\% CIs} \]

\[ \text{95\% CIs} \]

\[ \text{Sprint Number} \]

\[ \text{Sprint Number} \]
Conclusions

Summary, Contribution, Future Work and Publications
Practical and operational method to implement Agile MBSE

sMBSAP Meetings → Sprint Retro

sMBSAP Artifacts
- Physical Viewpoint
- Increment / Demo
- Operational Viewpoint
- Daily Standups

sMBSAP Roles
- Product Owner
- Scrum Master
- System Architect
- Development Team

sMBSAP Processes
- Initiate
- Plan and Architect
- Implement
- Review and Retrospect
- Release

Modified to integrate Scrum and MBSAP
Inherited from MBSAP
Inherited from Scrum
Common in both Scrum and MBSAP
## Summary (2 of 2)

<table>
<thead>
<tr>
<th>Metric</th>
<th>Scrum</th>
<th>sMBSAP</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commitment Reliability</td>
<td>0.81</td>
<td>0.94</td>
<td>sMBSAP has a statistically significant larger mean</td>
</tr>
<tr>
<td>Velocity</td>
<td>26.8</td>
<td>30.4</td>
<td>sMBSAP has a statistically significant larger mean</td>
</tr>
<tr>
<td>Velocity Fluctuation</td>
<td>0.08</td>
<td>0.07</td>
<td>No considerable difference</td>
</tr>
<tr>
<td>CLOC per Hour</td>
<td>58.03</td>
<td>71.1</td>
<td>sMBSAP has a numerically larger ratio</td>
</tr>
<tr>
<td>Defect Density (PBIs)</td>
<td>0.91</td>
<td>0.62</td>
<td>sMBSAP has a statistically significant smaller mean</td>
</tr>
<tr>
<td>Defect Density (KLOC)</td>
<td>4.57</td>
<td>2.24</td>
<td>sMBSAP has a numerically smaller ratio</td>
</tr>
<tr>
<td>Defect Leakage</td>
<td>19.6%</td>
<td>15.4%</td>
<td>sMBSAP has a statistically significant smaller mean</td>
</tr>
</tbody>
</table>
A practical and operational method for merging Agile and MBSE validated through architecting and developing a prototype for a health tech system. In parallel, the results suggest that sMBSAP is a middle ground hat is more aligned with federal and state regulations, as it addresses the technical debt concerns.

Research Contribution

- **sMBSAP Meetings**
  - **Sprint / Architecture Review**
  - **Product/Sprint Backlog**
- **sMBSAP Artifacts**
  - **Physical Viewpoint**
  - **Operational Viewpoint**
- **sMBSAP Roles**
  - Product Owner
  - Scrum Master
  - System Architect
  - Development Team
- **sMBSAP Processes**
  - **Initiate**
    - **Inputs**
      - Project Business Case
    - **Processes**
      - Create Project and Product Scope
      - Identify Project Stakeholders and Project Team
      - Create Architecture Overview and Summary
      - Create Product Breakdown
      - Create Product Backlog
      - Develop Release Plan
    - **Outputs**
      - Personas
      - Prioritized Product Backlog
      - System Architecture Overview and Summary
      - Release Plan
      - Core Project Team
  - **Plan and Architect**
    - **Inputs**
      - Personas
      - Prioritized Product Backlog
      - System Architecture Overview and Summary
      - Release Plan
      - Core Project Team
    - **Processes**
      - Create and Update Backing Items
      - Create System Architecture
      - Create and Update System Architecture
      - Create Product Breakdown
      - Estimate Backing Items
    - **Outputs**
      - Acceptance criteria of the Product Backlog Items
      - System Architecture
      - Acceptance criteria of the System Architecture
      - System Architecture Backlog
      - System Architecture
  - **Implement**
    - **Inputs**
      - Acceptance criteria of the Product Backlog Items
      - System Architecture
      - Product Backlog Items
      - System Architecture Backlog
      - System Architecture
    - **Processes**
      - Create Deliverables/Product Increments
      - Communicate Progress
      - Create and Update System Architecture
      - Develop System Architecture
      - Update System Architecture
    - **Outputs**
      - Accepted System Architecture
      - Accepted Deliverables
      - List of Actionable Improvements
      - Deliver Architecture Models
      - Retrospected Project
      - Working Deliverables
      - Lessons Learned for Future Implementation
  - **Review and Retrospect**
    - **Processes**
      - Demonstrate and Validate Deliverables
      - Retrospect Sprint
    - **Outputs**
      - Accepted System Architecture
      - Accepted Deliverables
      - List of Actionable Improvements
      - Deliver Architecture Models
      - Retrospected Project
      - Working Deliverables
      - Lessons Learned for Future Implementation
Research Contribution

- A practical and operational method for merging Agile and MBSE validated through architecting and developing a prototype for a health tech system. In parallel, the results suggest that sMBSAP is a middle ground hat is more aligned with federal and state regulations, as it addresses the technical debt concerns.

- **Architecture framework and Reference Architecture** for describing a health tech system that defines the structure, data, behavior, and capabilities.
Research Contribution

- **A practical and operational method for merging Agile and MBSE** validated through architecting and developing a prototype for a health tech system. In parallel, the results suggest that sMBSAP is a middle ground hat is more aligned with federal and state regulations, as it addresses the technical debt concerns.

- **Architecture framework and Reference Architecture** for describing a health tech system that defines the structure, data, behavior, and capabilities

- **Experiment** providing **side-by-side quantitative comparisons of** an MBSE-based methodology and an Agile-based methodology to quantitatively measure the benefits of one over the other.
Future Work

• Developing the same product in two parallel tracks (i.e., developing the same product twice); one track using an Agile MBSE and the other using an Agile approach.
• Considering a true experimental design with random sampling from a given population when assigning an approach to a given sprint.
• Conducting more comparative analyses between Agile and Agile MBSE methods using other software development objectives, techniques, and metrics.
• Generating executable code from implementation-level models and its effect on productivity and reusability.
• Examining the influence of requirements volatility on the Defect Rate.
• Testing the sMBSAP methodology while considering the impact of soft or human-related factors on software development Productivity.
• Investigating the implementation of the sMBSAP methodology while considering the non-technical tasks consumed during the sprints would be an interesting extension.

Awards

- **Award 1**: Graduate Student Showcase 2020 – Drivers of Innovation: The CSU Ventures Award
- **Award 2**: CSU Demo Day 2021 - Innovation In Software Award
- **Award 3**: CSU Demo Day 2022 - Innovation In Software Award
- **Award 4 (declined)**: VPR Fellows Cohort of 2021-2022 ($4000)
Acknowledgements
References (1 of 3)


References (3 of 3)


Thank you