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

Colorado State University

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

1

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

SWITRS

Statewide Integrated Traffic Records System

NATIONAL OILWELL VARCO

- **Academic Background**
  - *Bachelor of Science, Industrial Engineering from Zagazig University (Egypt)*
  - *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

FIFA WORLD CUP Qatar2022

British Petroleum

TANGGUH LNG

≡ CAL MATTERS

State tech failures hit home again

BY DAN WALTERS
AUGUST 12, 2020

SOUTHERN CALIFORNIA

California Wastes Billions on Failing Tech Projects, Researcher Says

By Siyamak Khorrami and Vanessa Serna | December 8, 2020 Updated: December 14, 2020 Print

THE SACRAMENTO BEE

The big reason California state computer projects fail

Fixing California's Government Technology Failures

TECHNOLOGY

California is the world's tech capital, but state computers are failing residents

If California can't get government tech right, who can?

By DEBRA KAHN | 08/22/2020 11:08 AM EDT

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?

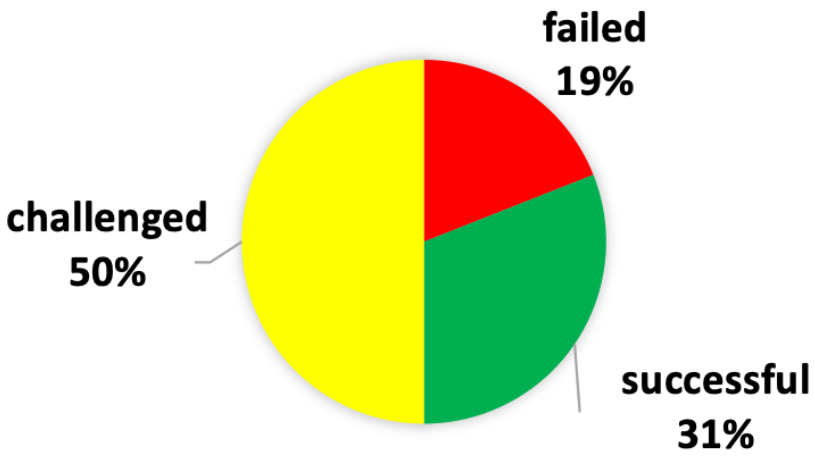
Bullet train to nowhere?

Challenges of Software Systems

%, projects >$15 million, in 2010 dollars

Project type	Average cost overrun	Average schedule overrun	Average benefits shortfall
Software	66	33	17
Nonsoftware	43	3.6	133
Total	45	7	56

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

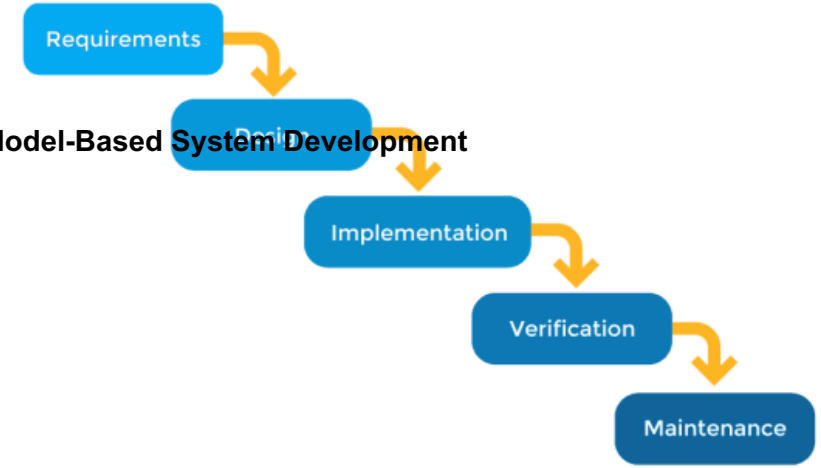


Standish Group CHAOS report 2020 [5]

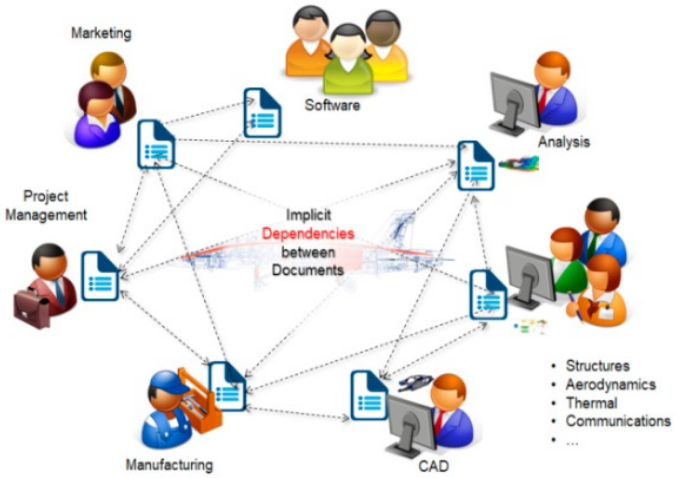
From DBSE to MBSE

Traditional System Development Methods

Waterfall: Software development activities are completed sequentially.



Model-Based System Development



Document-Based System Development

Traditional System Development Methods

Waterfall
V-Model
Spiral Model
RUP
RAD
Big Bang Model

Document—Based Systems Engineering (DBSE)

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

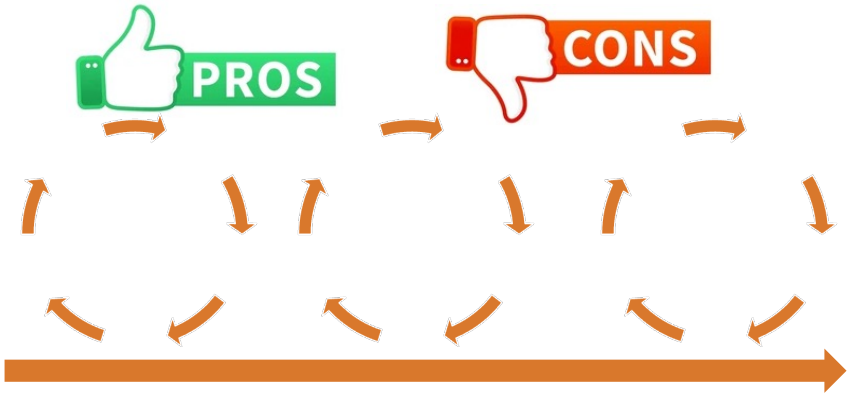
From DBSE to MBSE

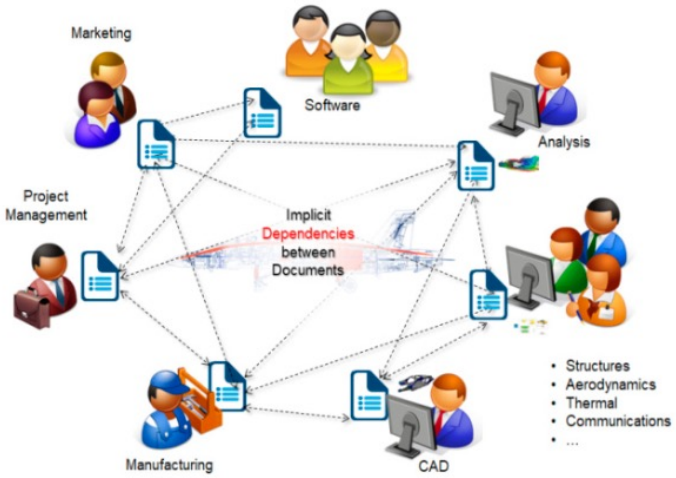
Traditional System Development Methods

Waterfall: Software development activities are completed sequentially.

Agile System Development Methods

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





Document-Based System Development

Agile System Development Methods

Agile Modeling
ASD
AUP
Crystal Clear
DSDM
FDD
Lean Software Development
SAFe
Scrum
XP
Rapid Prototyping
iterative Model
Incremental Model

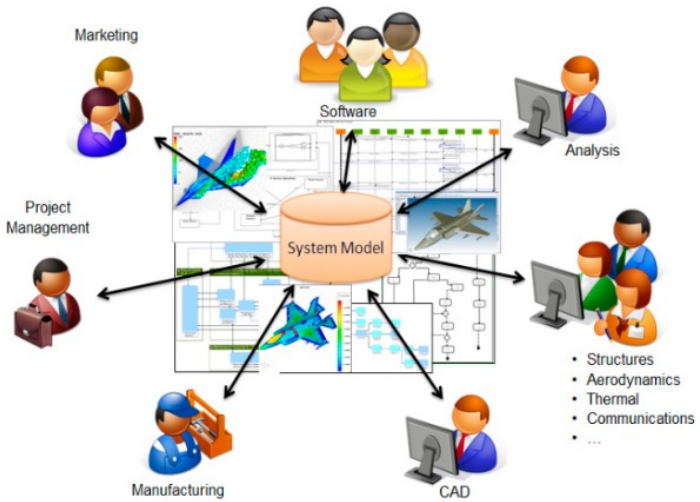
Traditional System Development Methods

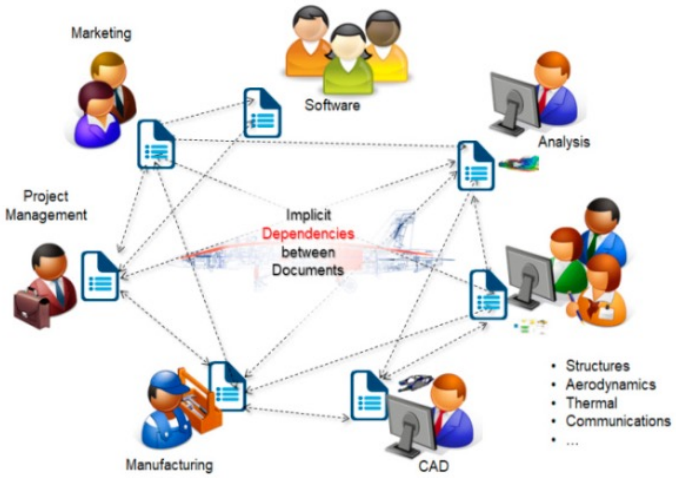
Waterfall
V-Model
Spiral Model
RUP
RAD
Big Bang Model

Document—Based Systems Engineering (DBSE)

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

From DBSE to MBSE





Model-Based System Development

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

Document-Based System Development

Agile System Development Methods

Agile Modeling
ASD
AUP
Crystal Clear
DSDM
FDD
Lean Software Development
SAFe
Scrum
XP
Rapid Prototyping
iterative Model
Incremental Model

Traditional System Development Methods

Waterfall
V-Model
Spiral Model
RUP
RAD
Big Bang Model

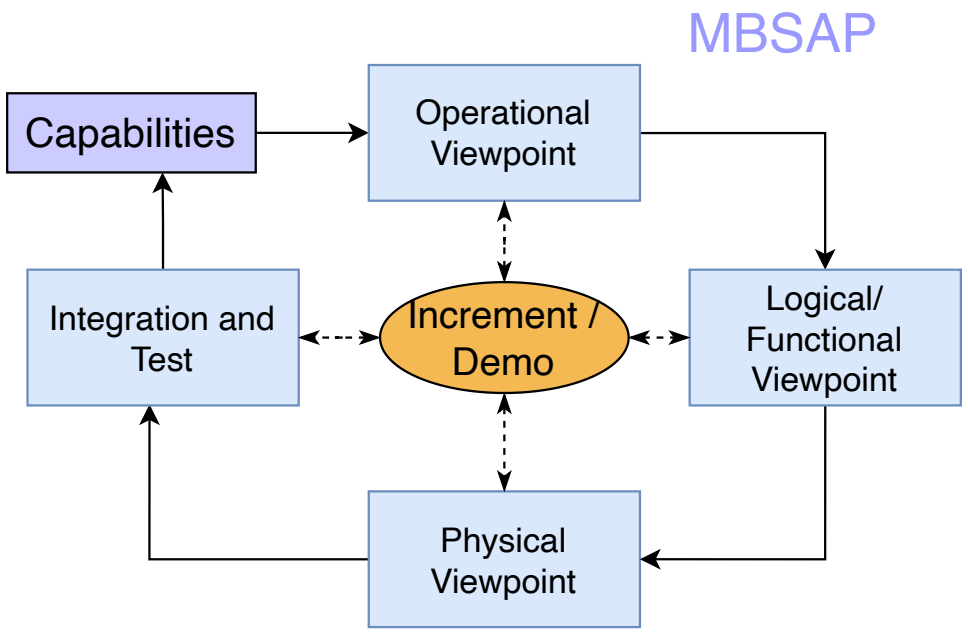
Model—Based Systems Engineering (MBSE)

Method to create an architecture for a system through structured decomposition into modular and manageable levels of complexity by using object-oriented principles.

Document—Based Systems Engineering (DBSE)

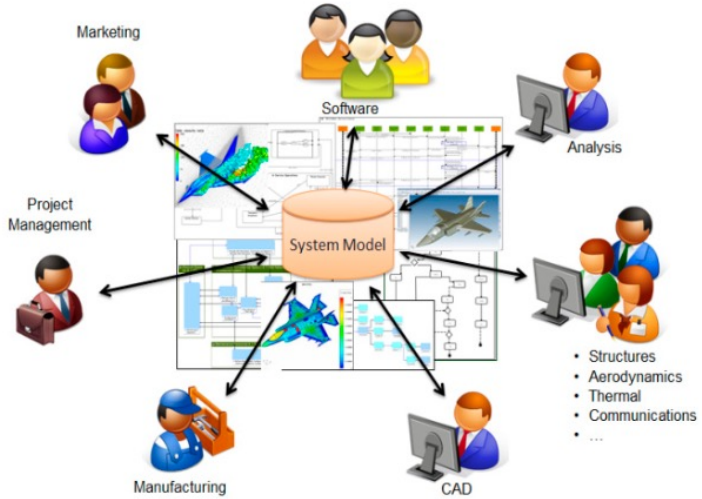
Methods that uses documents and multiple tools to capture and communicate the system design throughout the project lifecycle.

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 System Development

Architecture Frameworks

NIST Enterprise Architecture
TOGAF
FEAF
DoDAF
Zachman

MBSE Approach

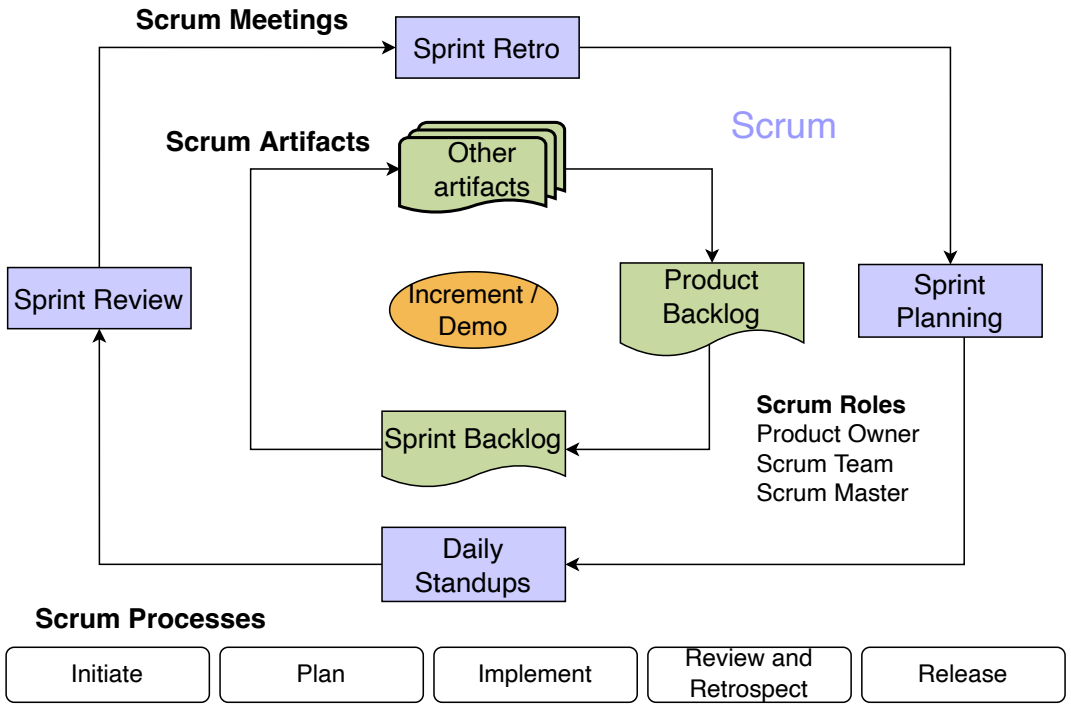
IBM Telelogic Harmony-SE
INCOSE OOSEM
IBM RUP SE for MDSD
Vitech MBSE
JPL State Analysis (SA)
Dori OPM
MBSAP

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.

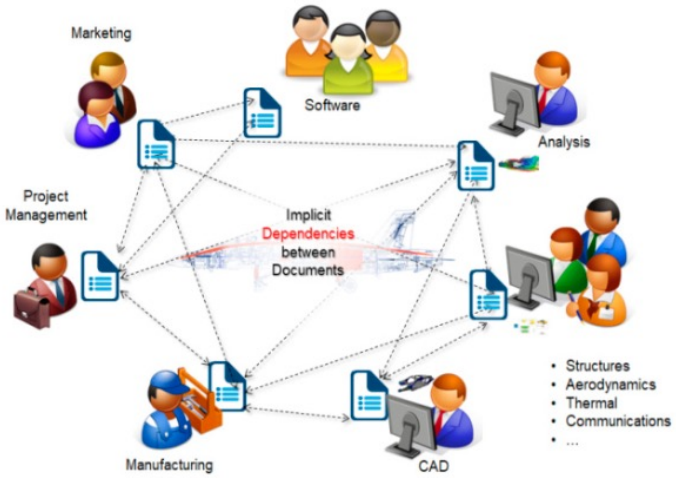
From Agile to Scrum

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



Scrum

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



Document-Based System Development

Agile System Development Methods

Agile Modeling
ASD
AUP
Crystal Clear
DSDM
FDD
Lean Software Development
SAFe
Scrum
XP
Rapid Prototyping
iterative Model
Incremental Model

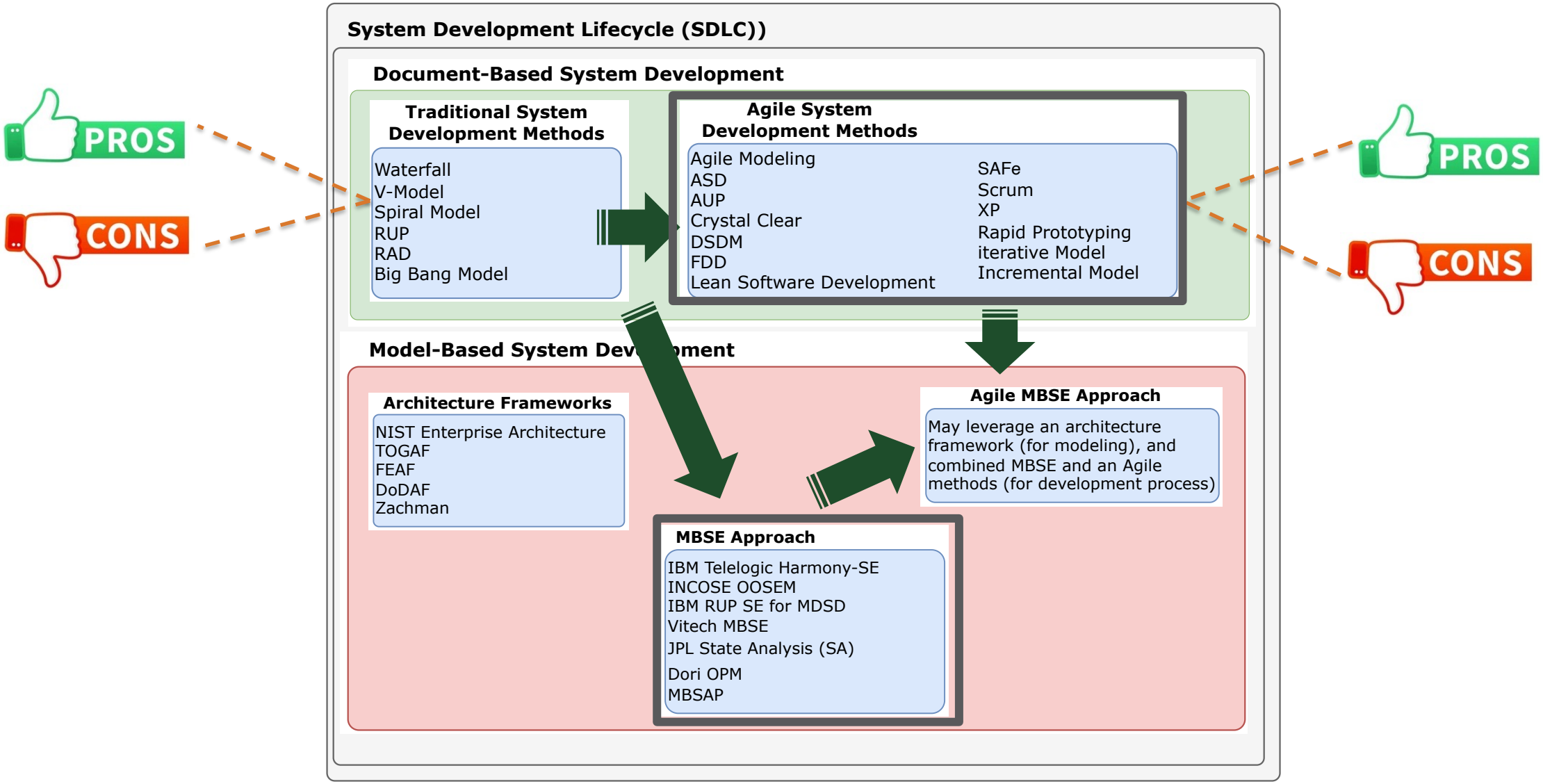
Traditional System Development Methods

Waterfall
V-Model
Spiral Model
RUP
RAD
Big Bang Model

Document—Based Systems Engineering (DBSE)

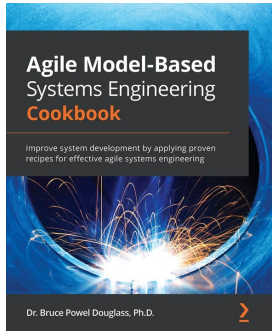
Methods that uses documents and multiple tools to capture and communicate the system design throughout the project lifecycle.

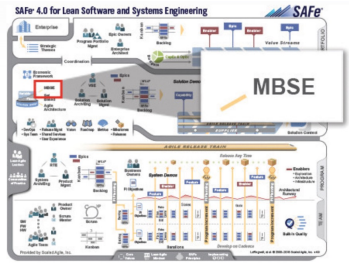
Towards Agile MBSE

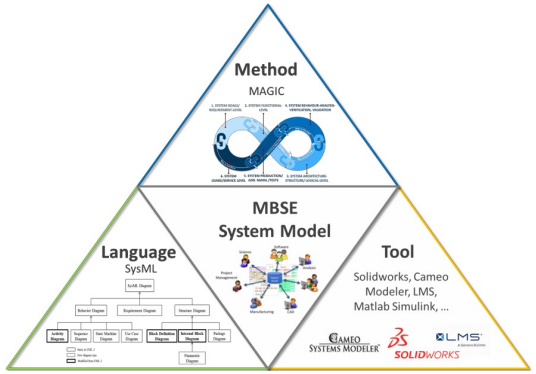


SDLC methods, pros and cons [1,18,22,25,38-58,203]

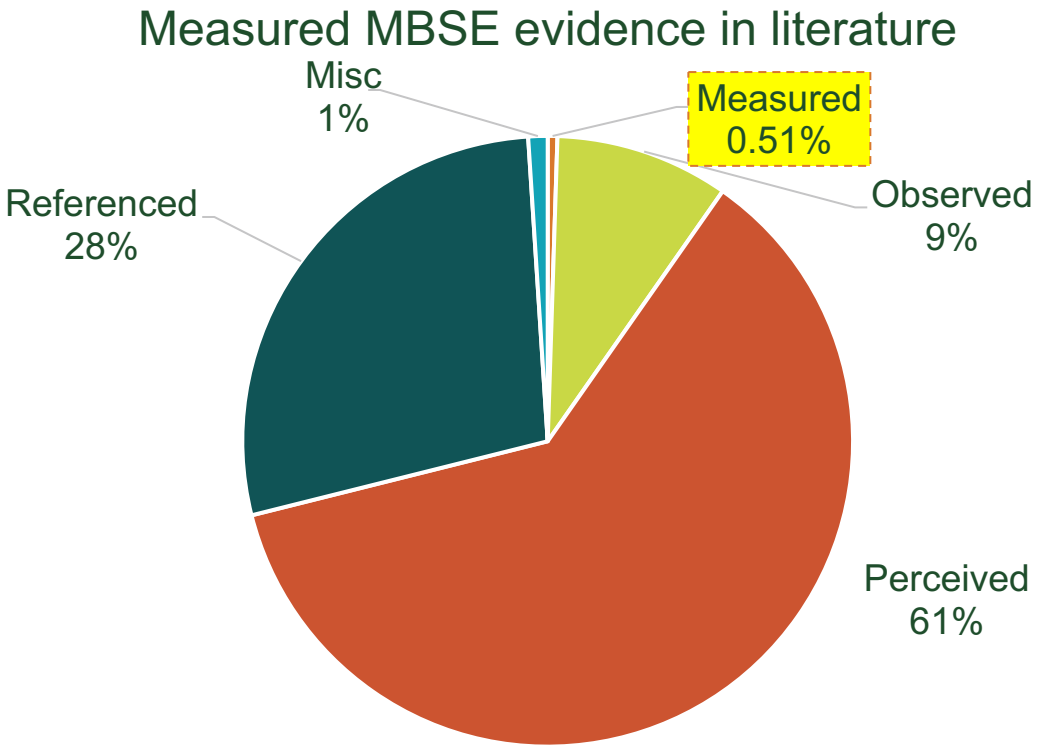
Research Gap







No practical guide found for implementing Agile MBSE for Software Systems [22-25], [204], [207].



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.

* Based on the review of 360 published research

Research Purpose (Stage 1)

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

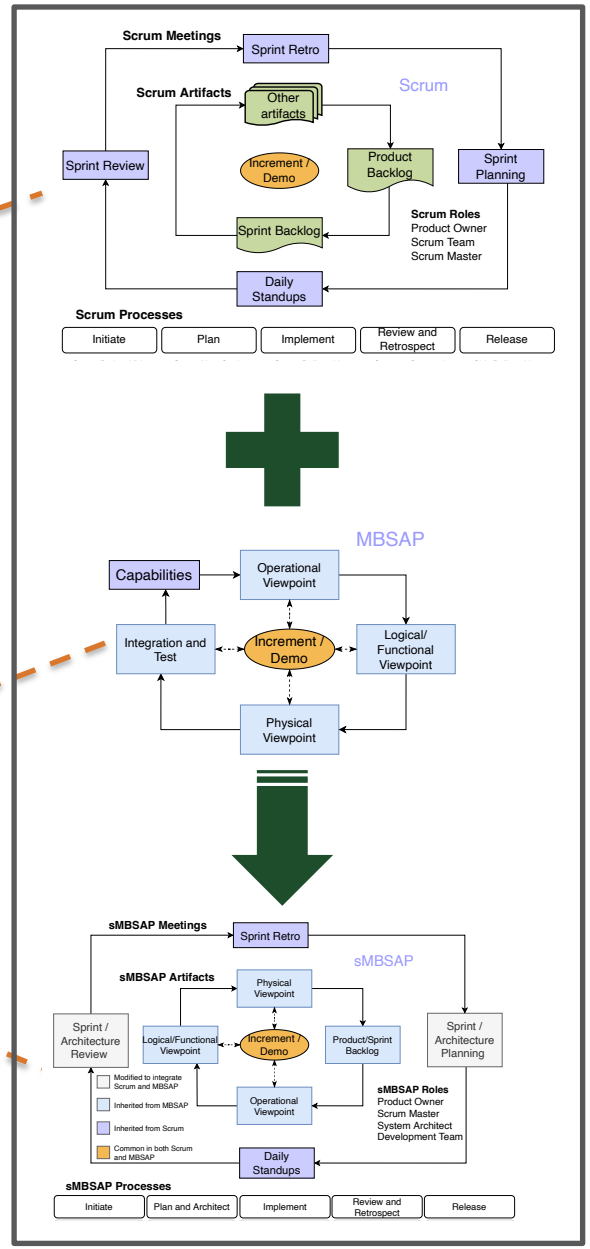
NIST Enterprise Architecture
TOGAF
FEAF
DoDAF
Zachman

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



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 H_0: 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 H_1: 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 H_0 : 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 H_1 : 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]*.

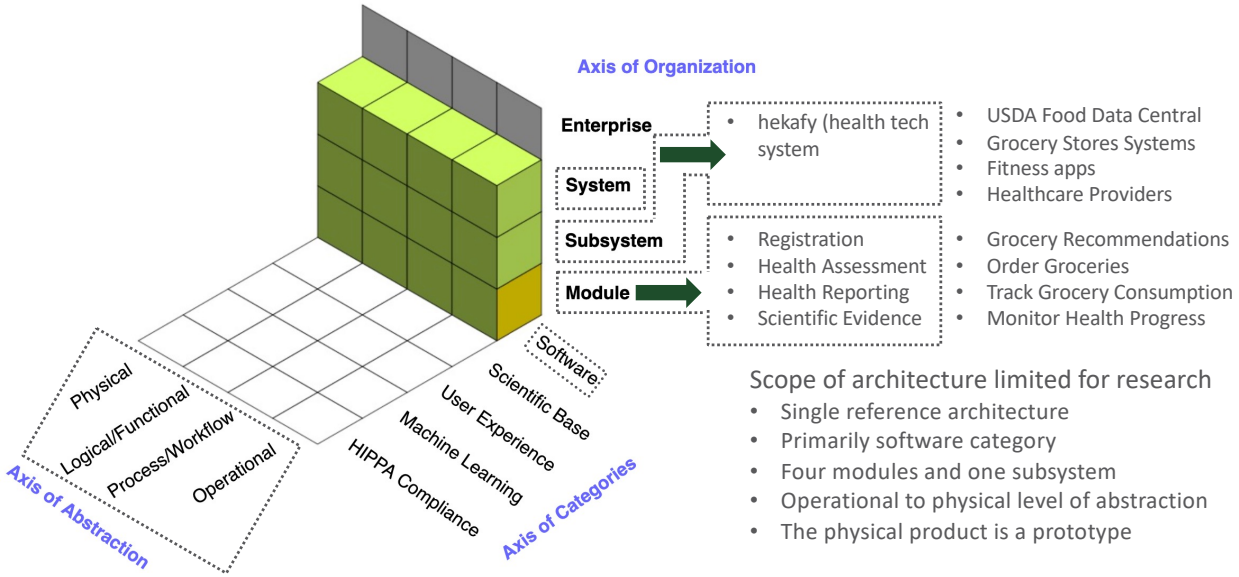
Third Hypothesis

Hypothesis H_0 : 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 H_1 : 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.

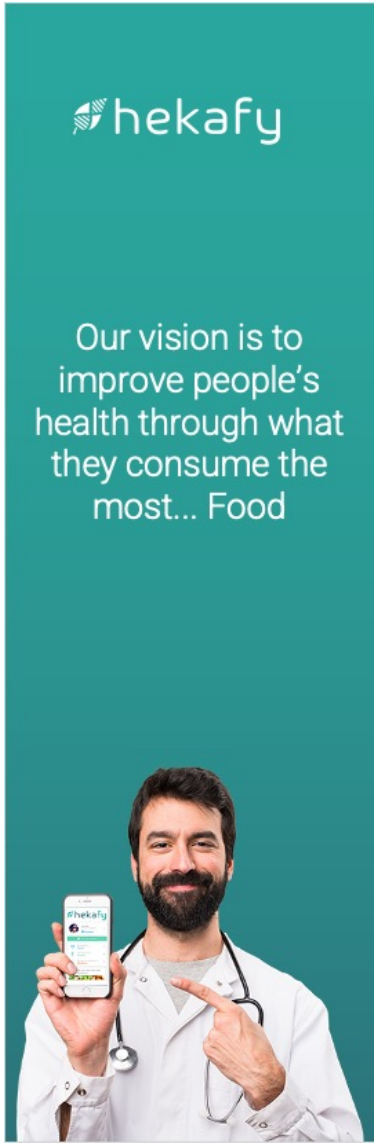
System Selected for the Experiment

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



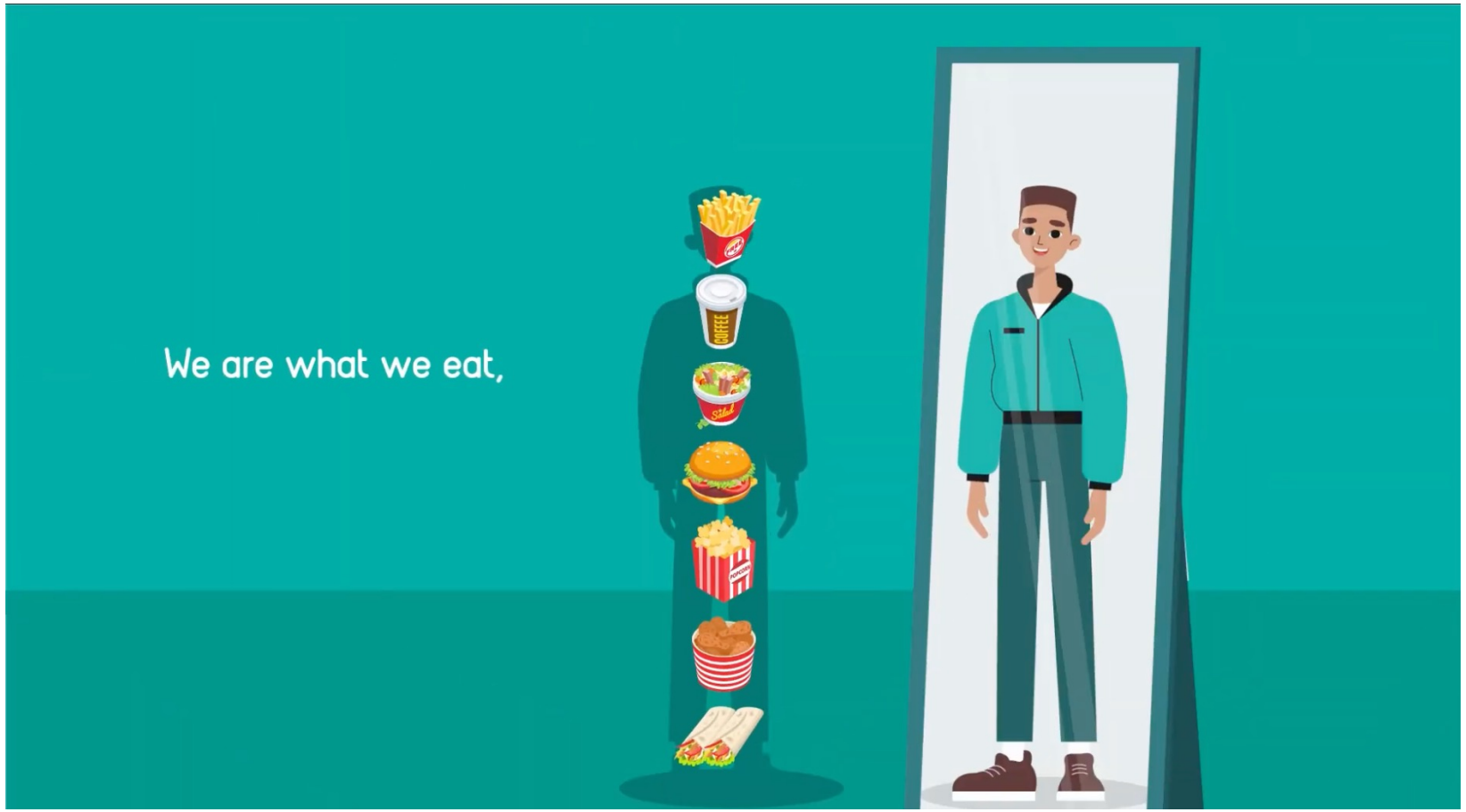
Scope of architecture limited for research

- 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



https://www.youtube.com/watch?v=mvDRgxgbYkM

2

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

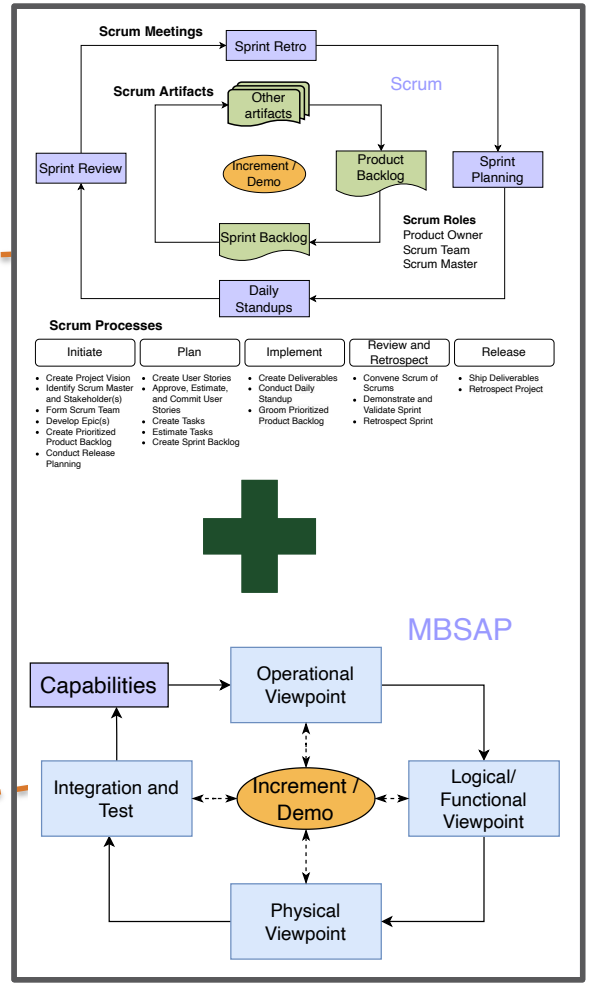
NIST Enterprise Architecture
TOGAF
FEAF
DoDAF
Zachman

Agile MBSE Approach

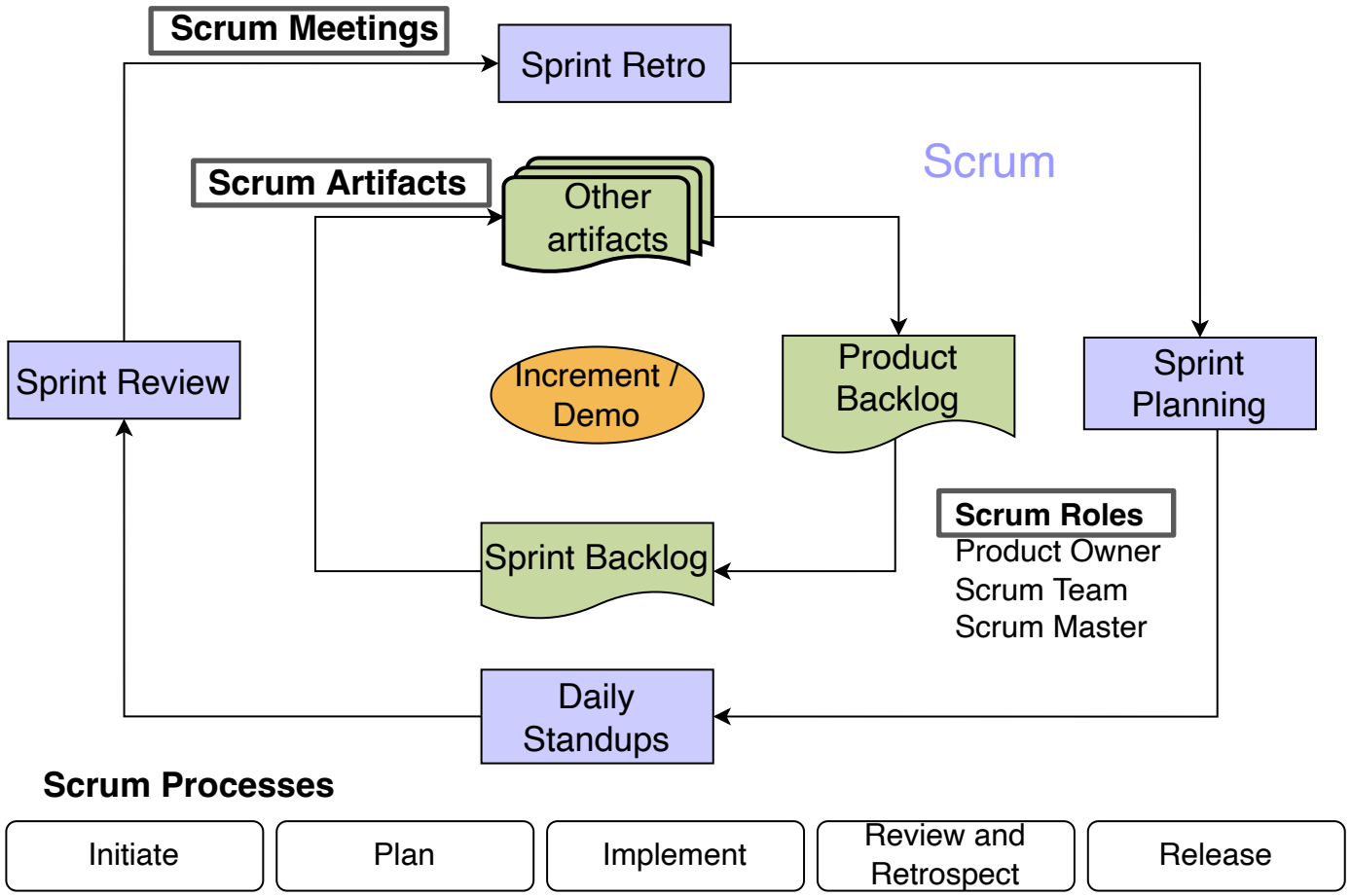
May leverage an architecture framework (for modeling), and combined MBSE and an Agile methods (for development process)

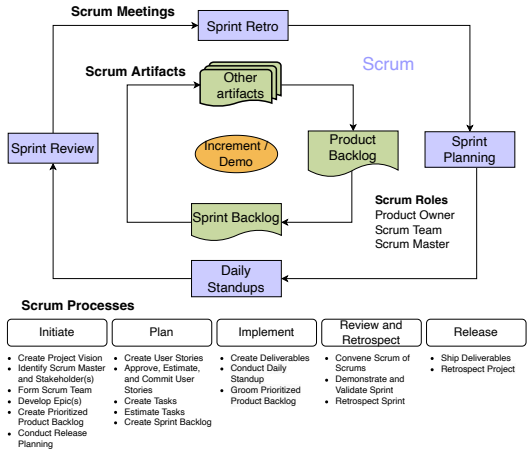
MBSE Approach

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

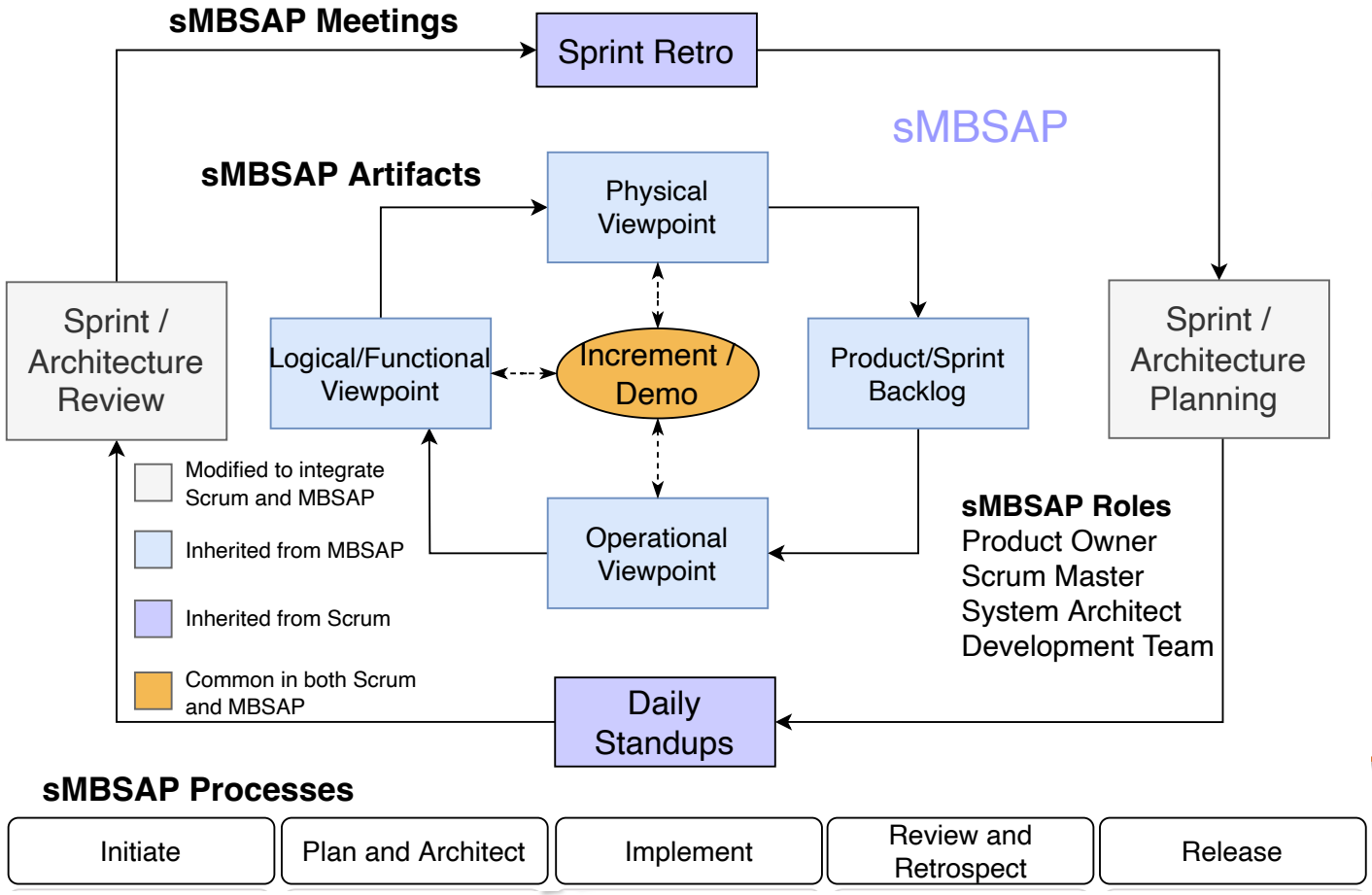


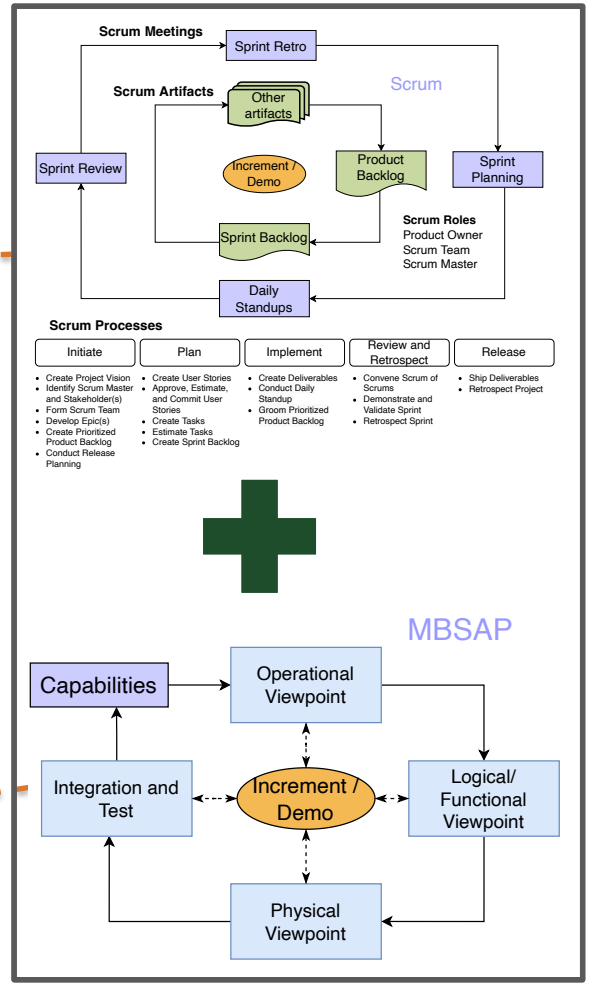
Agile (Scrum) + MBSE (MBSAP) = Agile MBSE (sMBSAP)



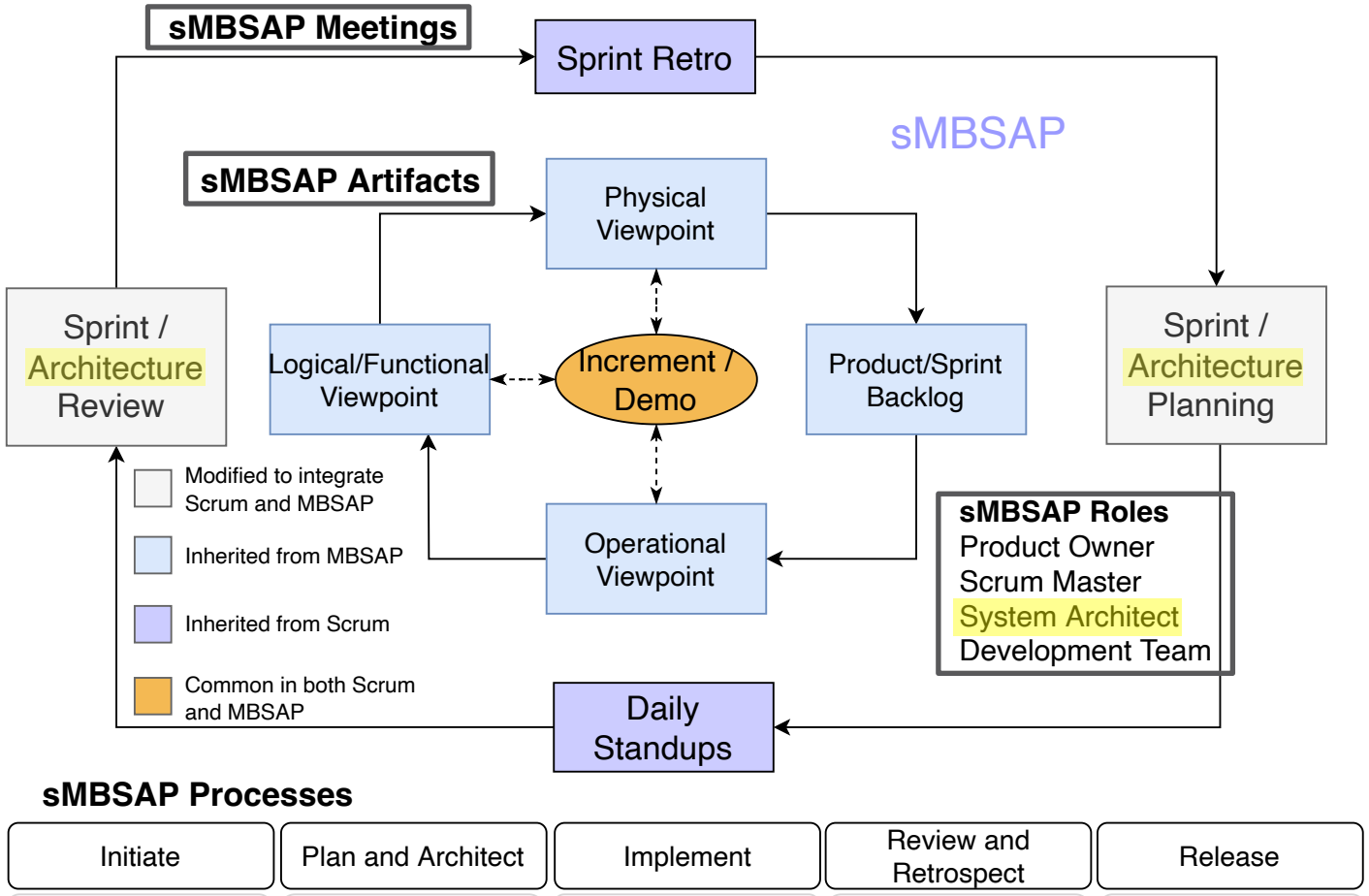


Agile (Scrum) + MBSE (MBSAP) = Agile MBSE (sMBSAP)





Agile (Scrum) + MBSE (MBSAP) = Agile MBSE (sMBSAP)



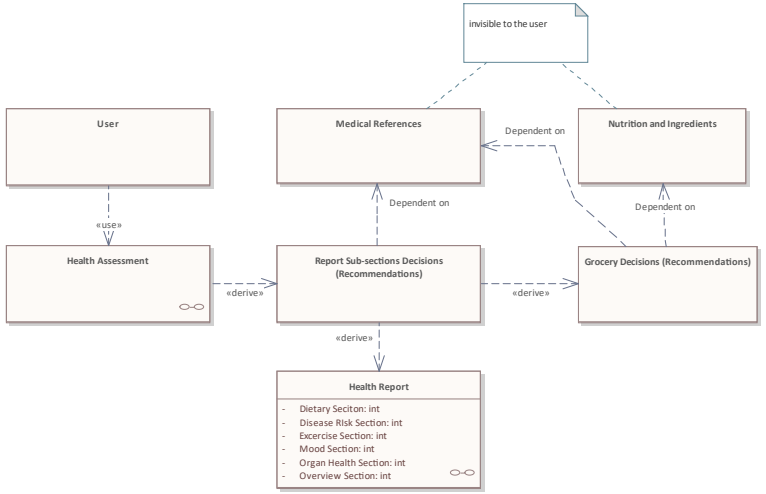
Engaging the team in customizing the tool

Modify the meetings to include architecture planning and review

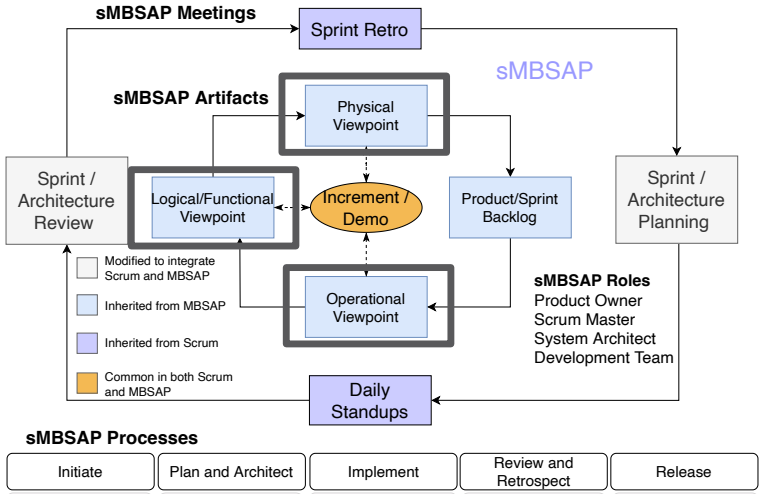
Using UML-based and non-UML-based models to describe the system

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].

Progression of Formal UML Models for the Health Tech System

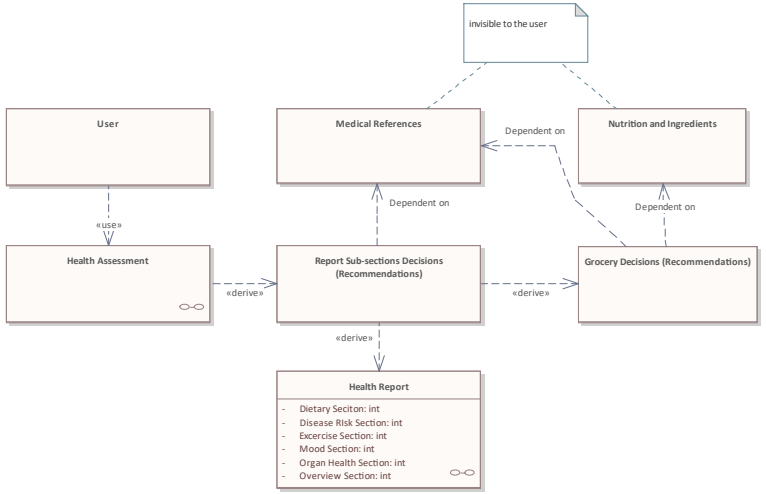


Conceptual Data Model



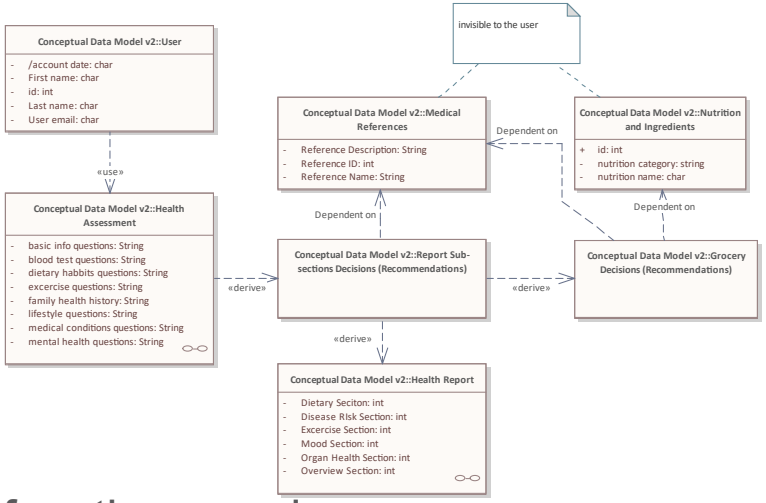
- Translate the Project Scope Statement, and Product Backlog into an architectural mode.
- A high-level view of the architecture.
- Define the system's boundary and context.
- Create top-level partitioning (Domains), primary Behaviors (Use Cases), and primary data content.

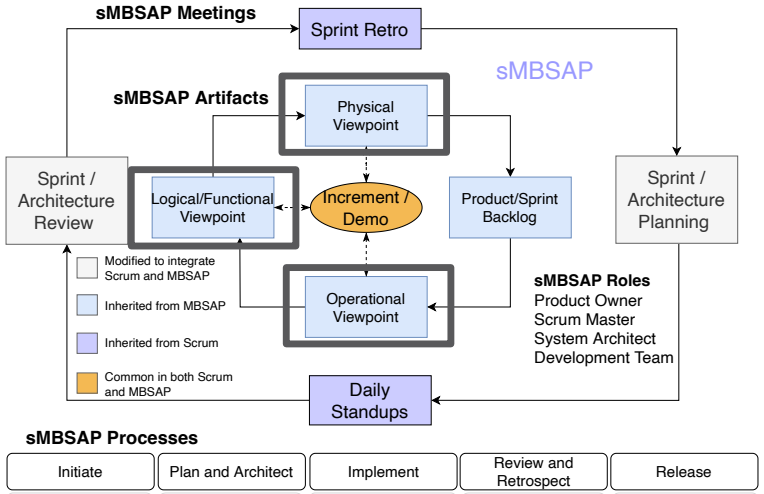
Progression of Formal UML Models for the Health Tech System



Conceptual Data Model

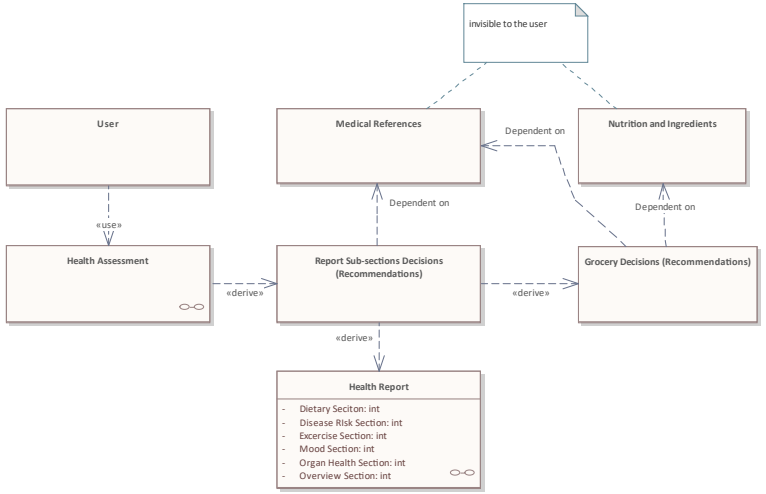
Logical Data Model





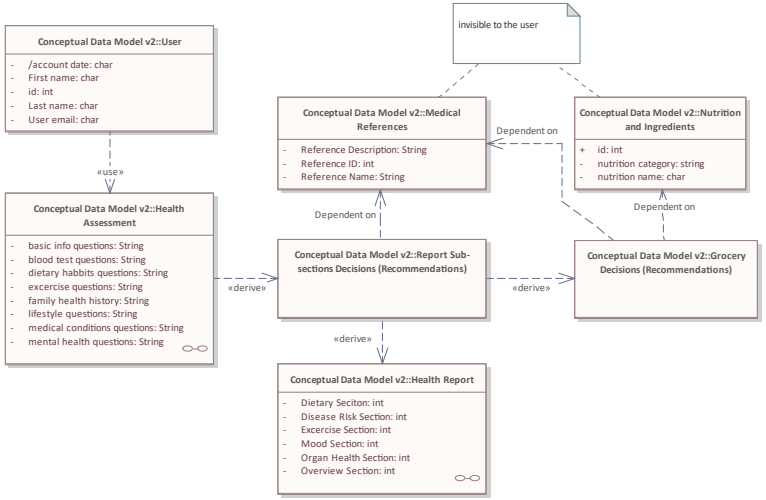
- Define the details of system elements, functions, and data.
- Progressive elaboration on the perspectives of the OV.
- Functional service specifications are developed and allocated to logical components and interfaces.
- Functional definition of the technology- and product-agnostic system.

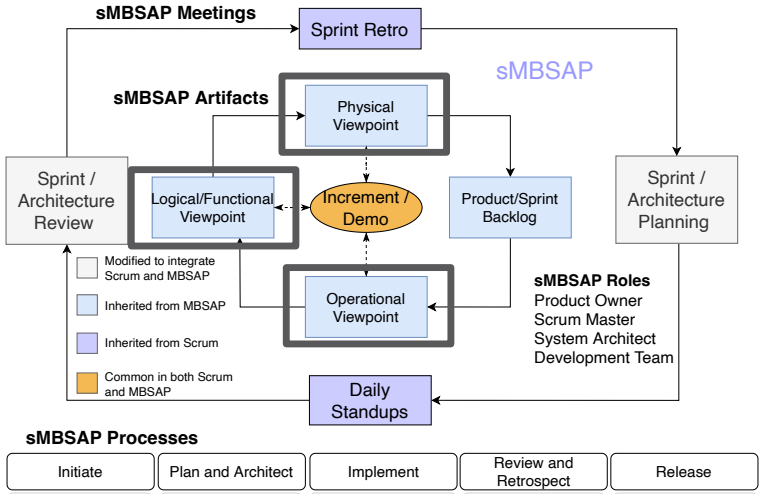
Progression of Formal UML Models for the Health Tech System

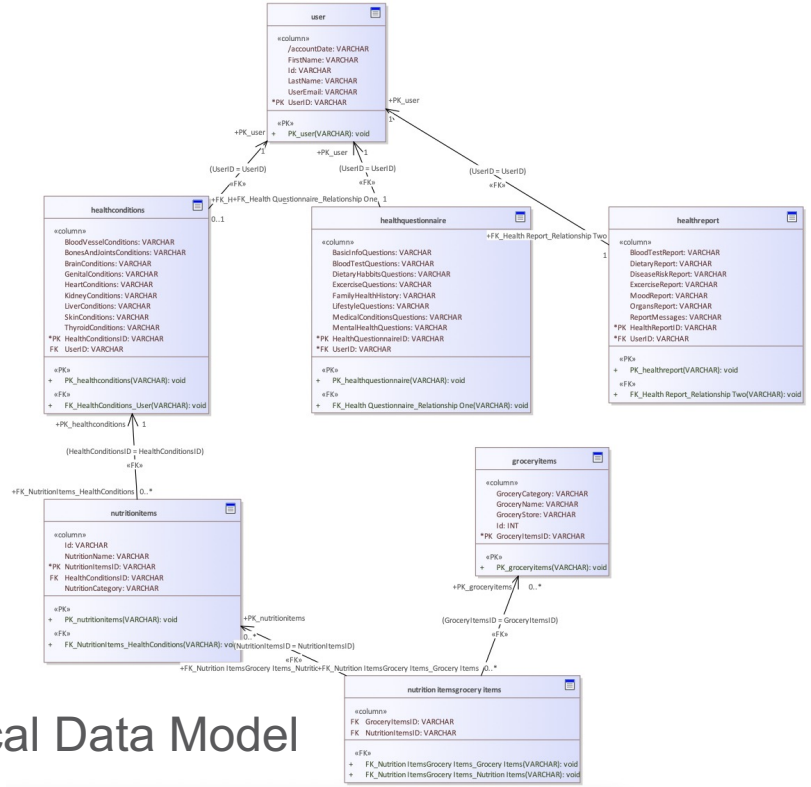


Conceptual Data Model

Logical Data Model



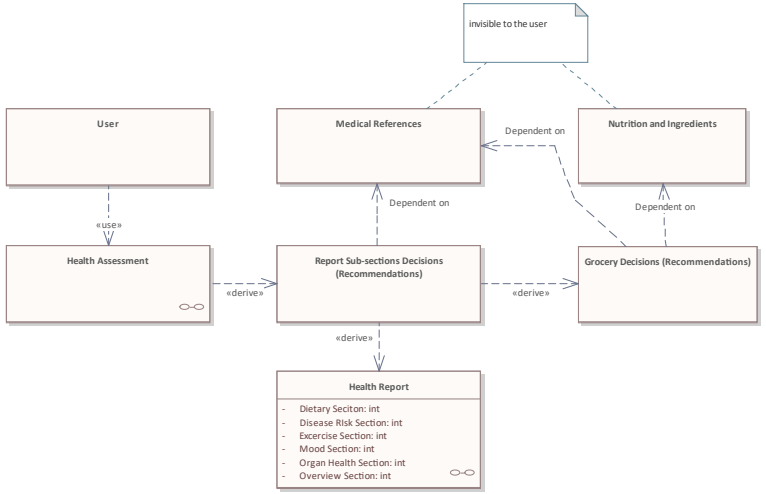




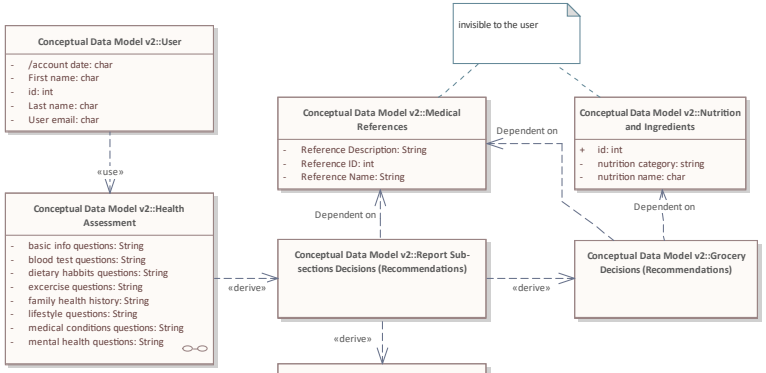
Physical Data Model

- 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.

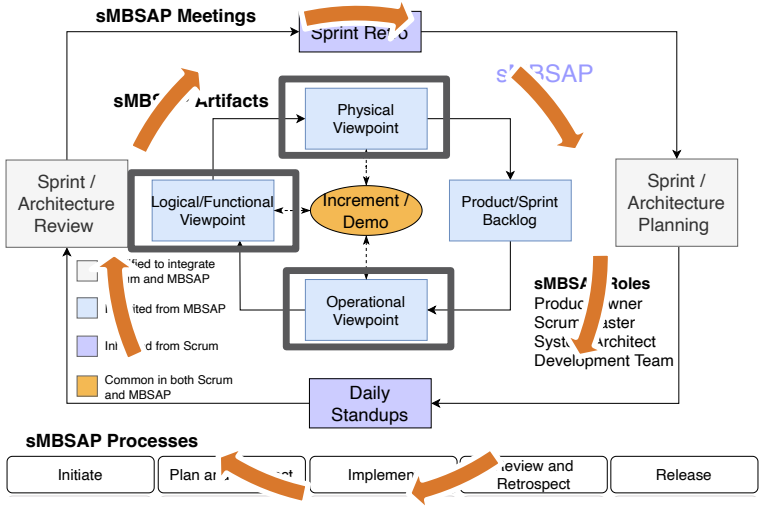
Progression of Formal UML Models for the Health Tech System

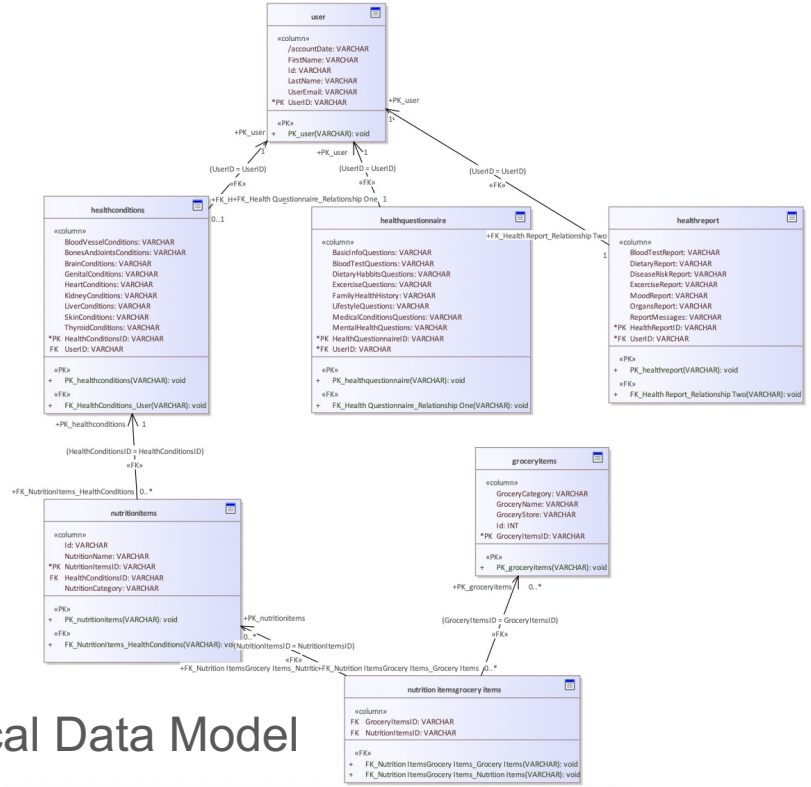


Conceptual Data Model



Logical Data Model

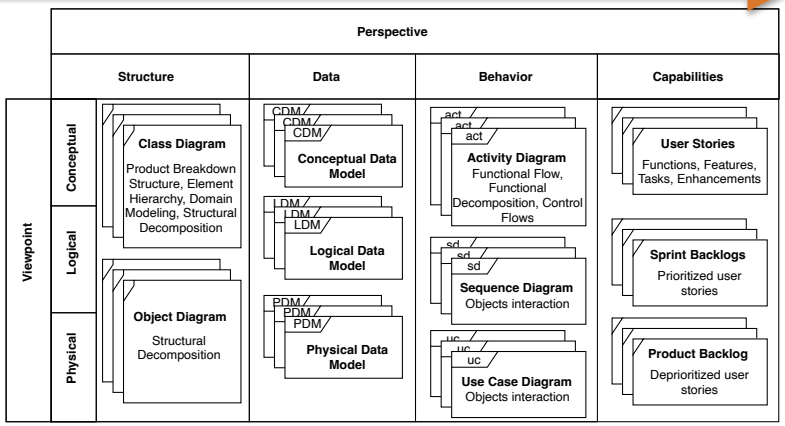




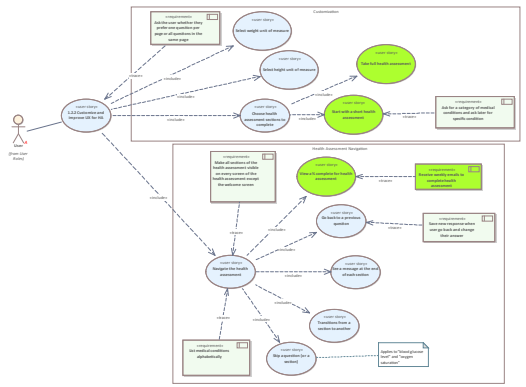
Physical Data Model

Perspectives (Structure, Data, Behavior, Capabilities)

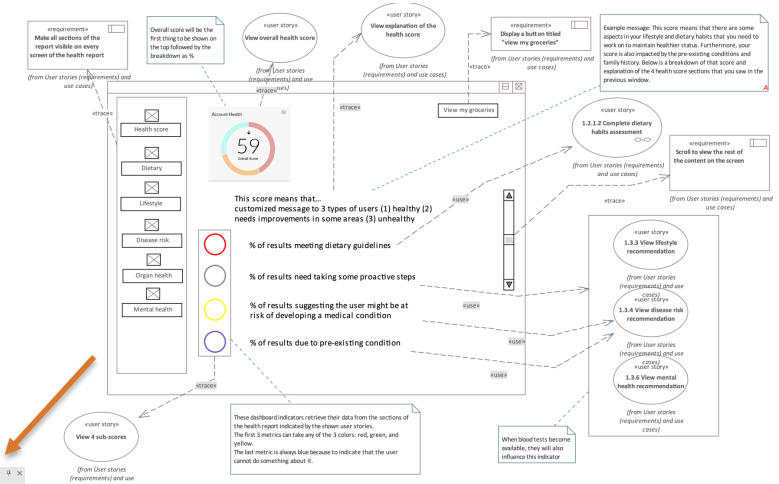
Viewpoints (Operational, Logical, and Physical)



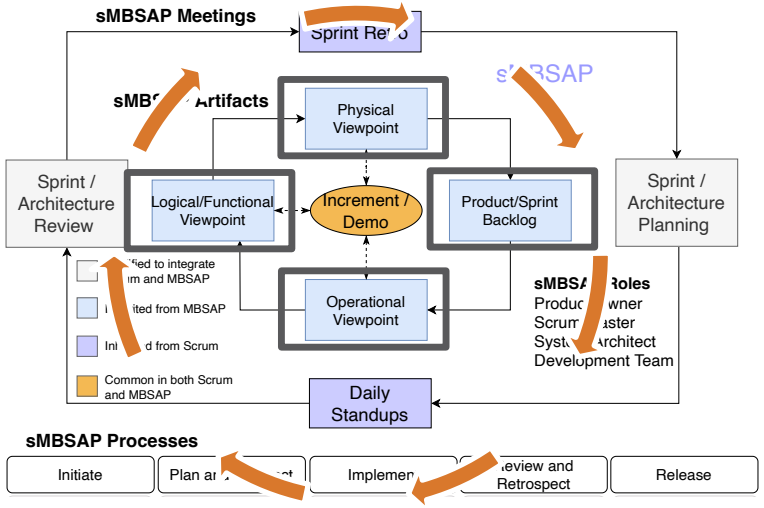
Integrating Informal Artifacts of the Health Tech System into the Model

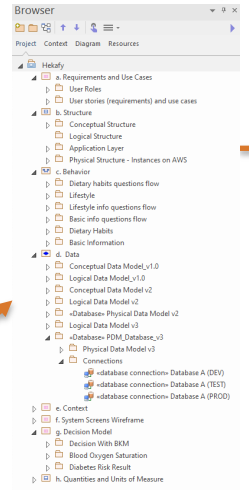


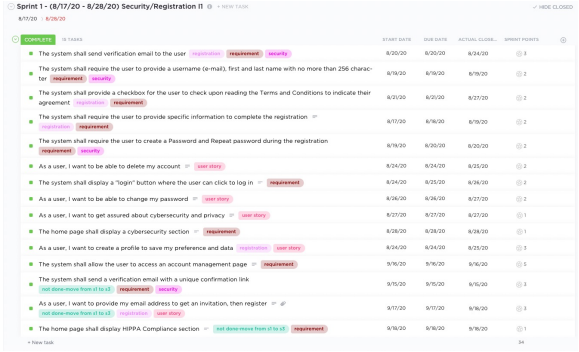
Visual Use Case diagrams which also includes user stories and requirements



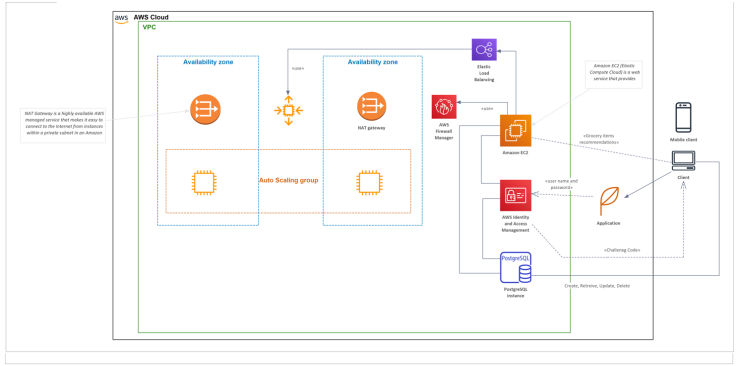
Wireframes (illustration of a screen's interface) integrated into the model and mapped to requirements







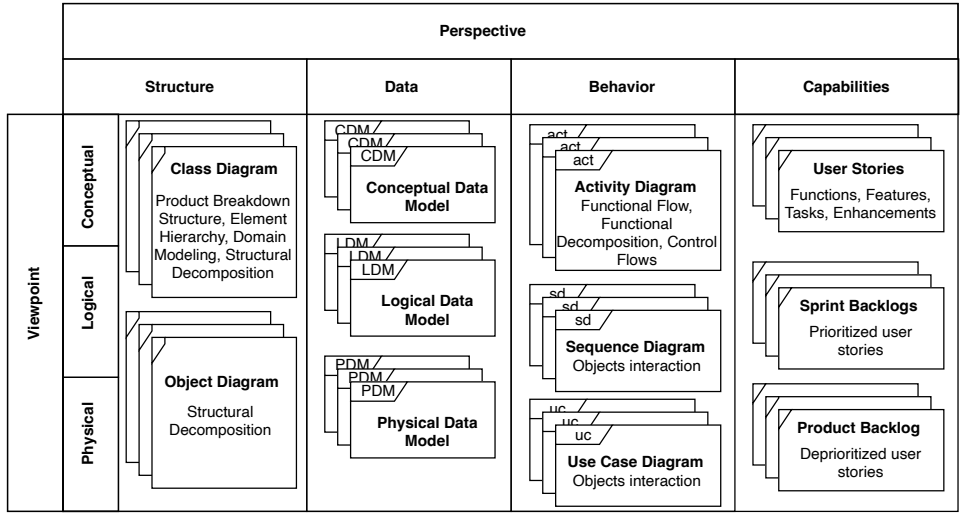
Text-based sprint log (user stories and requirements)



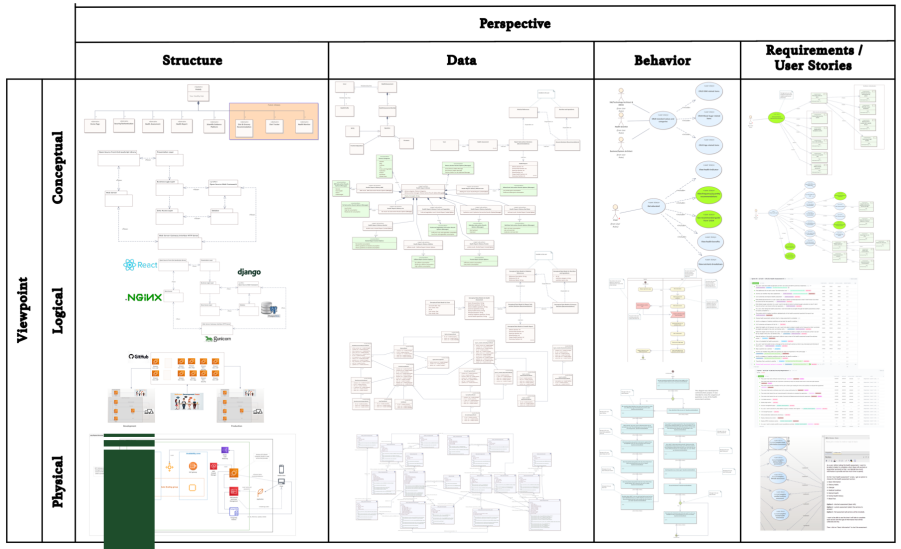
AWS Architecture Model

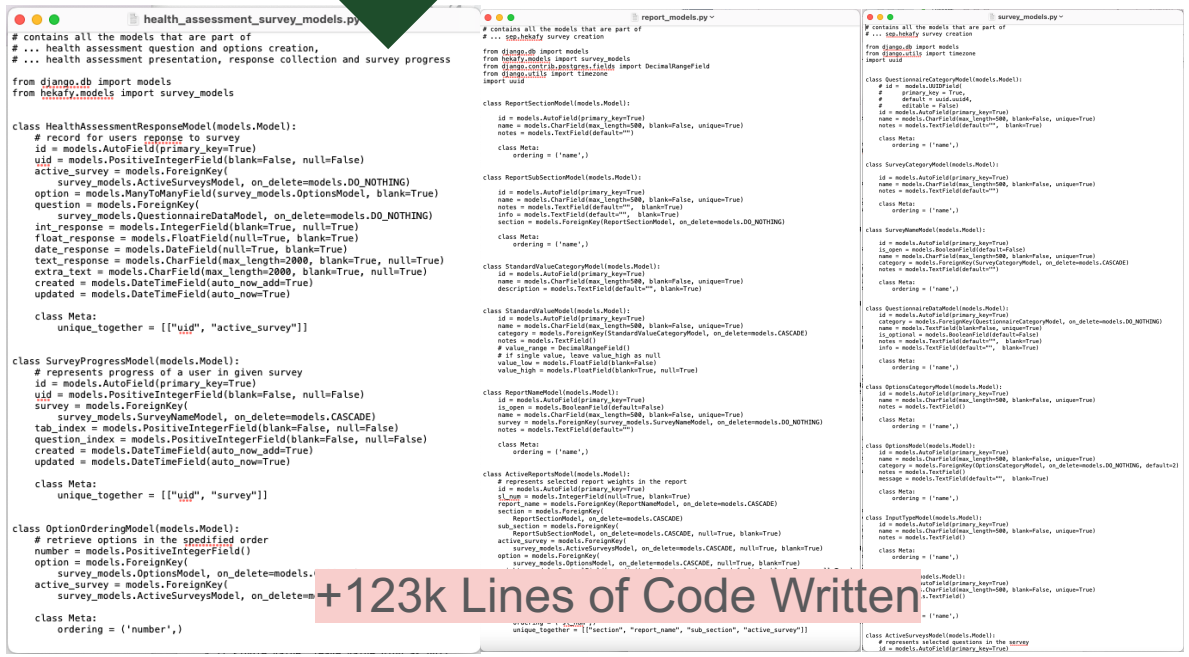
From sMBSAP artifacts to products

Architecture Framework



Synthesis of Artifacts



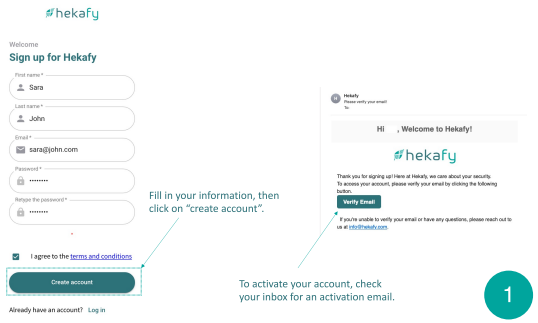


+123k Lines of Code Written

AWS Deployment

Name	Instance ID	Instance state	Instance type
sep.heakfy.co...	i-0bc018c9f1e6e4a84	Running	t2.small
login.hekafy....	i-00f7e4839a78f7b06	Running	t2.micro
web_dev_alp...	i-0b78fad171eab1d2e	Running	t2.micro
health.hekaf...	i-0c348829188175572	Running	t2.micro
v0.1_mobile_...	i-0c01f5eb0e5ae8134	Running	t2.micro
web_product...	i-027cc2ba1061cd389	Running	t2.micro
farid_wordpr...	i-024d406473de82bbd	Running	t2.micro
wp.hekafy	i-0ecd0850bd7d55b68	Running	t2.micro

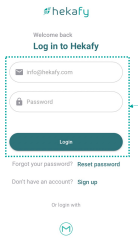
From sMBSAP artifacts to products



Fill in your information, then click on "create account".

To activate your account, check your inbox for an activation email.

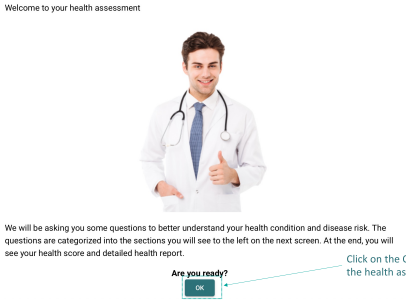
1



After you verify your email, you will be directed to the login page.

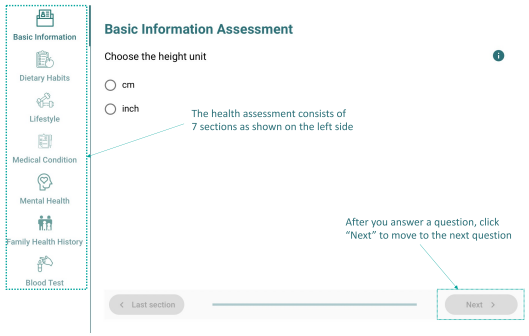
Enter you email and password and click "Login".

1



Click on the Ok button to start the health assessment

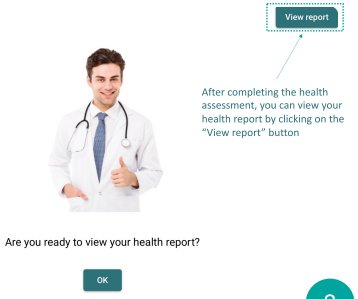
2



The health assessment consists of 7 sections as shown on the left side

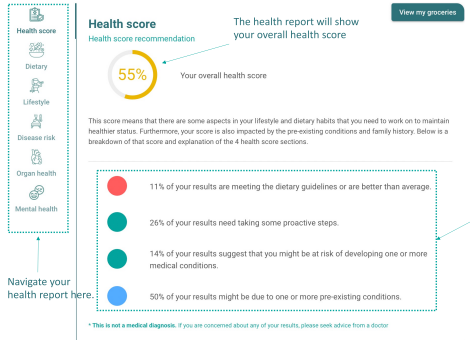
After you answer a question, click "Next" to move to the next question

2



After completing the health assessment, you can view your health report by clicking on the "View report" button

3



The health report will show your overall health score

Your high-level health indications are here

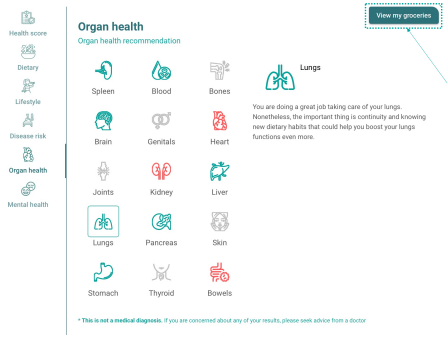
Navigate your health report here.

3

This feature is under development, and the data provided is for illustration purpose only.

We would love to hear from you. Click on the "Leave Feedback" button to tell us what you think.

4



Click here to explore groceries options that fit your health needs and goals.

3

Product Demo

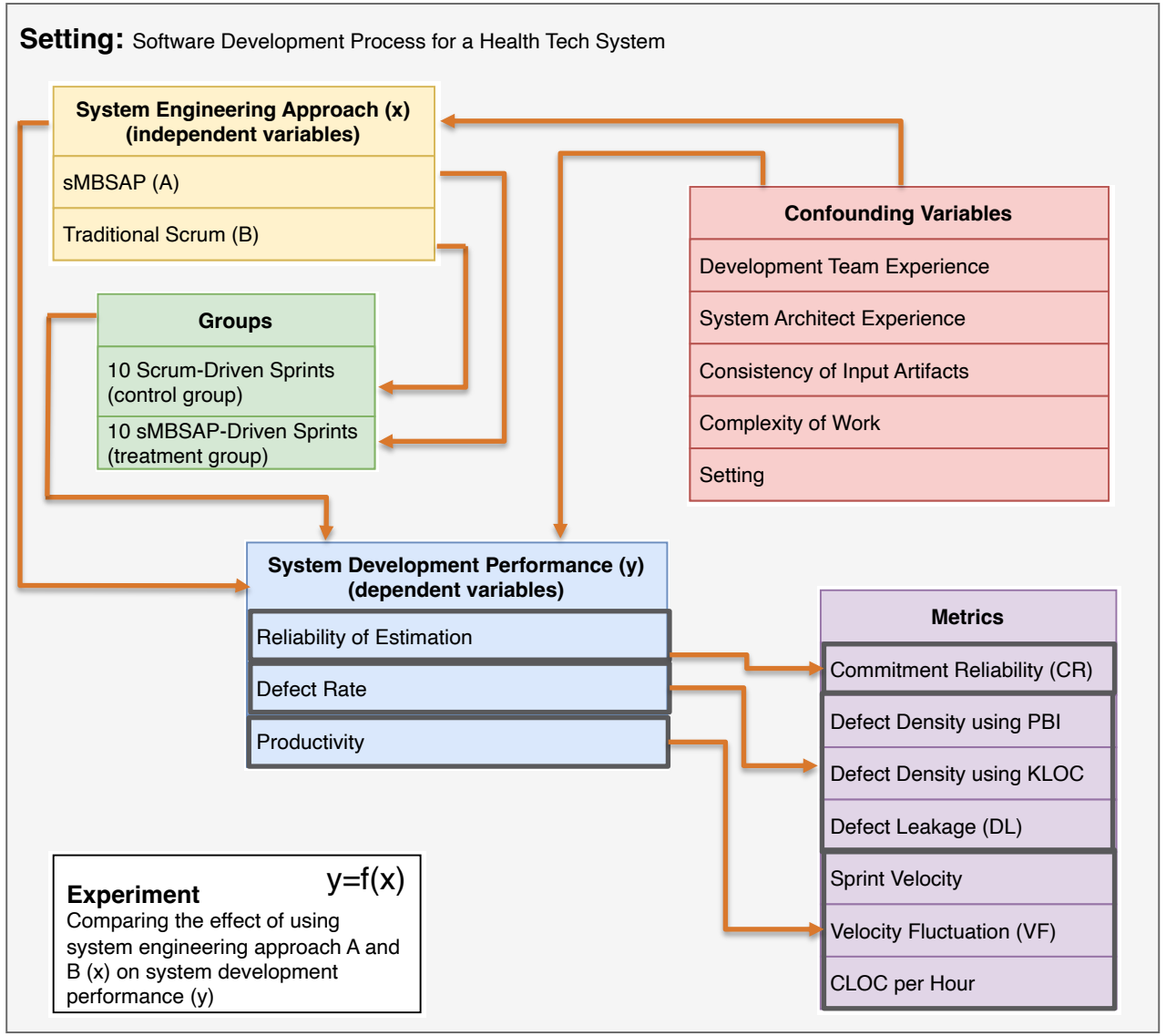
https://www.youtube.com/watch?v=L0X3ha2TeDY

3

Research Methodology for an Experiment to Compare sMBSAP and Agile | Stage 2

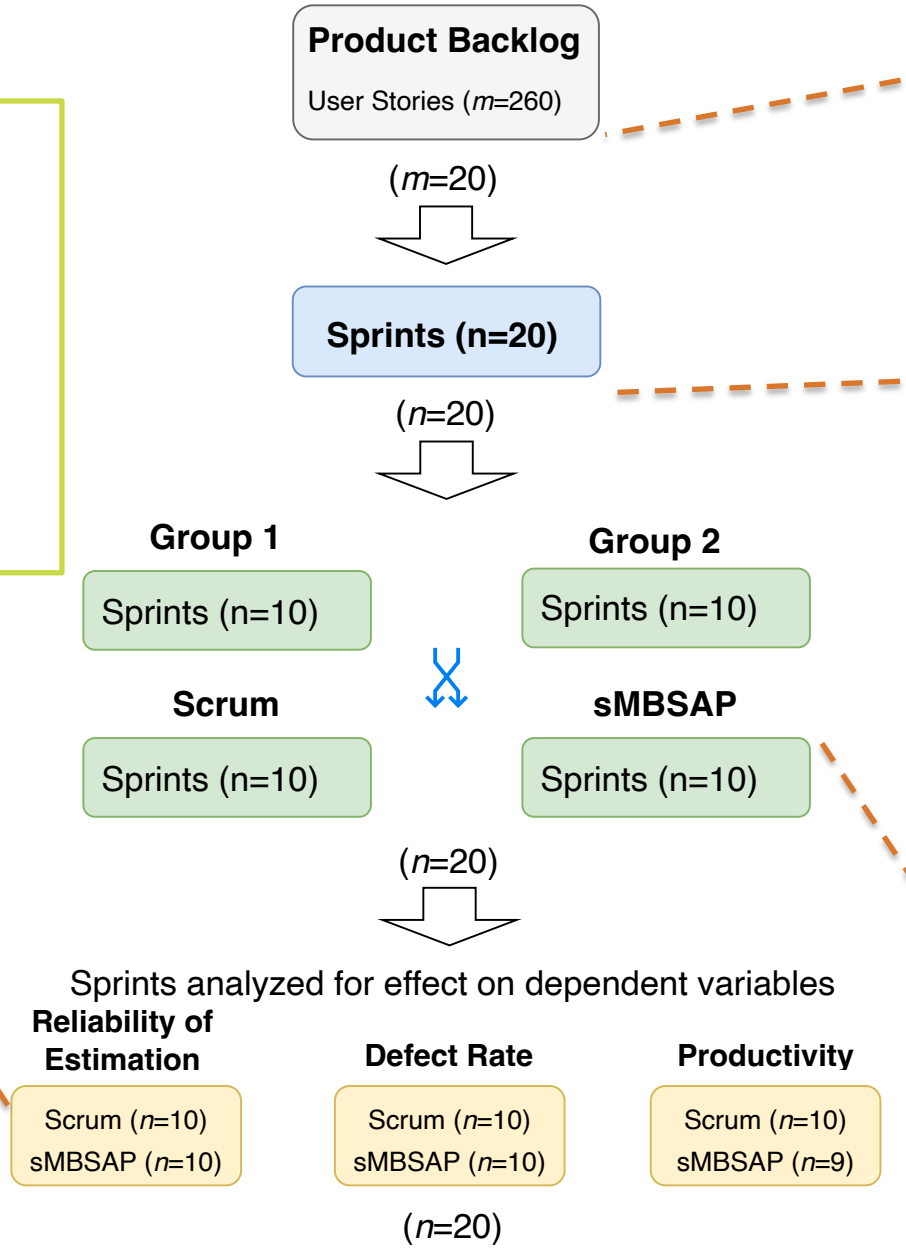
- Experiment Overview
- Experiment Design
- Techniques and Procedures
- Quality of Research Design

Experiment Overview



Experiment Design

④ 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)



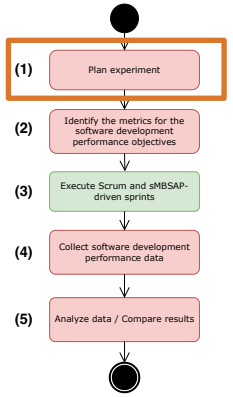
① All user stories (m=260) are identified and added to a product backlog

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).

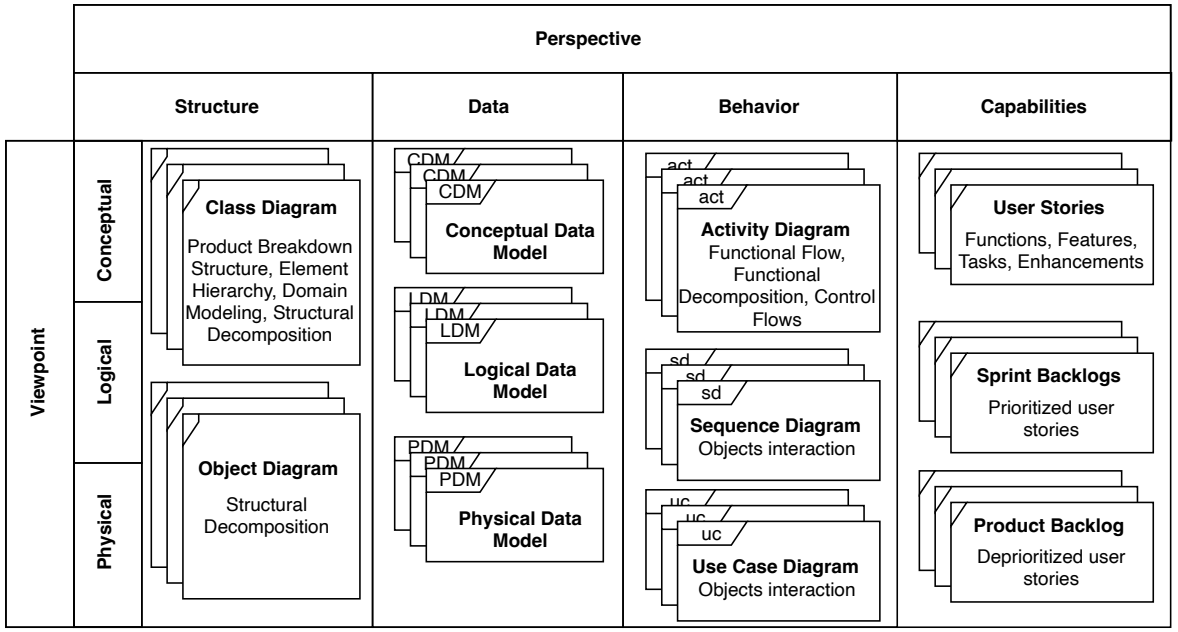
Techniques and Procedures (1 of 5)

① Plan Experiment

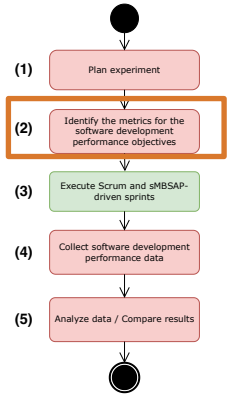


		Perspective			
		Structure	**Data**	**Behavior**	**Capabilities**
Viewpoint	**Operational** **Functional** **Physical**	**Classifications:** Hierarchy Levels **Product Breakdown:** Product Breakdown Structure, Structural Decompositions **Class Diagrams:** Domain Diagram with Specifications **Object Diagrams:** Decomposition of Domains to the level of Classes	**Classifications:** Hierarchy Levels **Conceptual Data Model (CDM):** Primary Data Entities, Foundation Classes **Logical Data Model (LDM):** Modeling Foundation Classes, Attributes and Operations **Physical Data Model (PDM):** Database Structure	**Use Case (UC) Diagram:** with Specifications **Activity Diagram (AD):** with multiple Domains and Actors **Sequence Diagram (SD):** Modeling non-stateful behavior of a group of classes	**Classifications:** Hierarchy Levels, User Story Types **Requirements:** Requirements Hierarchy, Requirement IDs, Requirements Texts, Verification Methods, Parents, Requirements Allocations

		Perspective			
		Structure	**Data**	**Behavior**	**Capabilities**
Viewpoint	**Operational** **Functional** **Physical**	Product Breakdown: Initiatives, Epics, and Stories	Data Model Slides	User Stories Flowcharts	User Stories (US), Product Backlog, Sprint Backlog



Techniques and Procedures (2 of 5)



② Identify the Metrics for the Performance of Software Development

Metrics for Reliability of Estimation

$$CR = \frac{Completed\ SP}{Planned\ SP}$$

$$Average\ CR = \frac{\sum CR}{Number\ of\ Sprints}$$

Metrics for Productivity

$$SV = \sum Completed\ SP\ During\ a\ Sprint$$

$$Average\ Velocity = \frac{\sum Sprint\ Velocity}{Number\ of\ Sprints}$$

$$VF = \frac{Average\ Velocity - Sprint\ Velocity}{Average\ Velocity}$$

$$CLOC\ per\ hour = \frac{CLOC}{Development\ Duration\ (hrs)}$$

Metrics for Defect Rate

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

$$Average\ DD = \frac{\sum DD}{Number\ of\ Sprints}$$

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

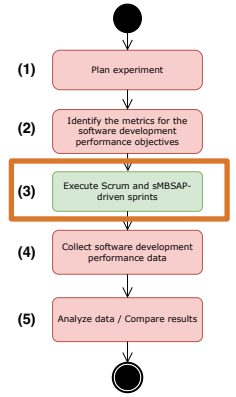
$$DL = \frac{Defect\ Count\ \text{(post-delivery)}}{Defect\ Count\ \text{(pre-delivery)}}$$

$$Average\ DL = \frac{\sum DL}{Number\ of\ Sprints}$$

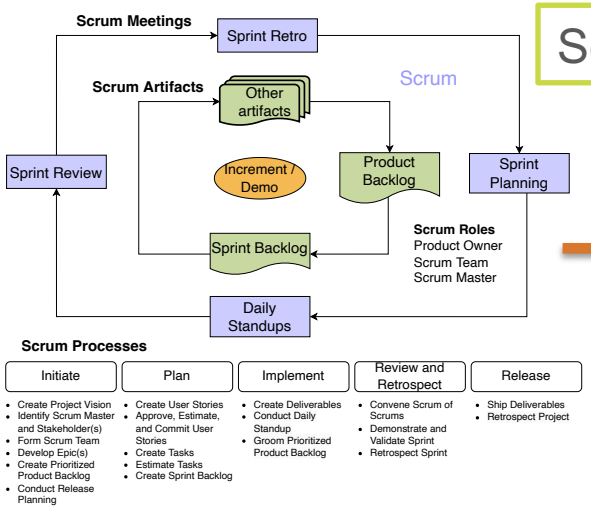
Metrics for Reliability of Estimation [74-77,99]; Metrics for Productivity [99,109,103-105], and Defect Rate [64,116-118,120,122,125]

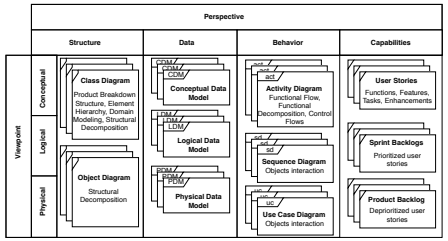
Techniques and Procedures (3 of 5)

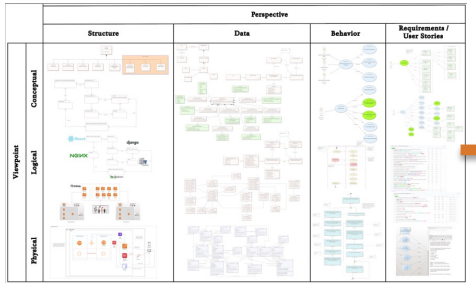
③ Execute Scrum and sMBSAP Drive-Sprints



Scrum





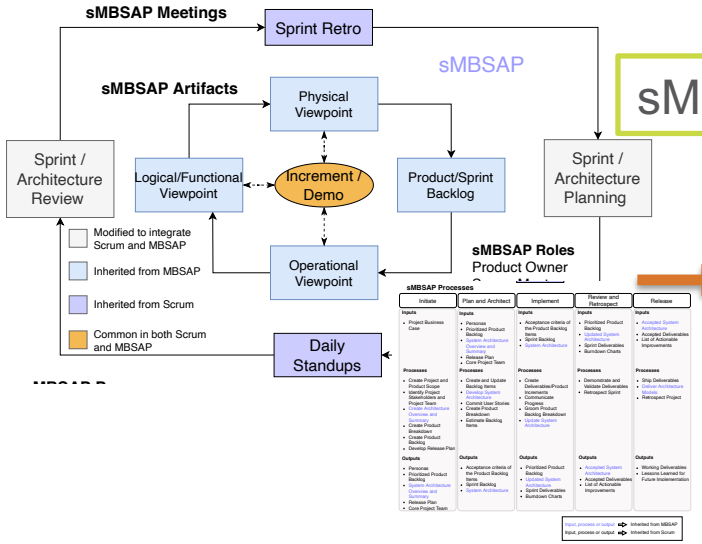


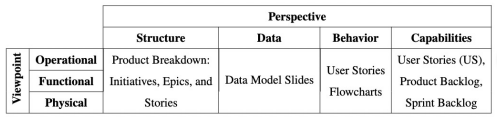
Architecture/Artifacts framework

Develop artifacts and implementing user stories

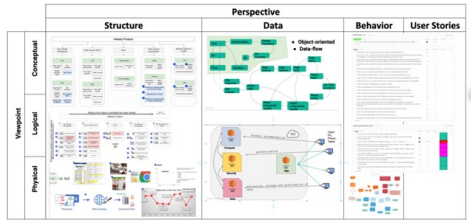
Write code

sMBSAP



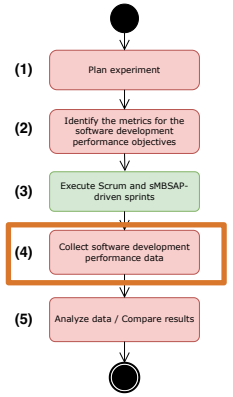


Viewpoint	Perspective			
	Structure	Data	Behavior	Capabilities
Operational	Product Breakdown: Initiatives, Epics, and Stories	Data Model Slides	User Stories Flowcharts	User Stories (US), Product Backlog, Sprint Backlog
Functional				
Physical				



Techniques and Procedures (4 of 5)

④ Collect System Development Performance Data



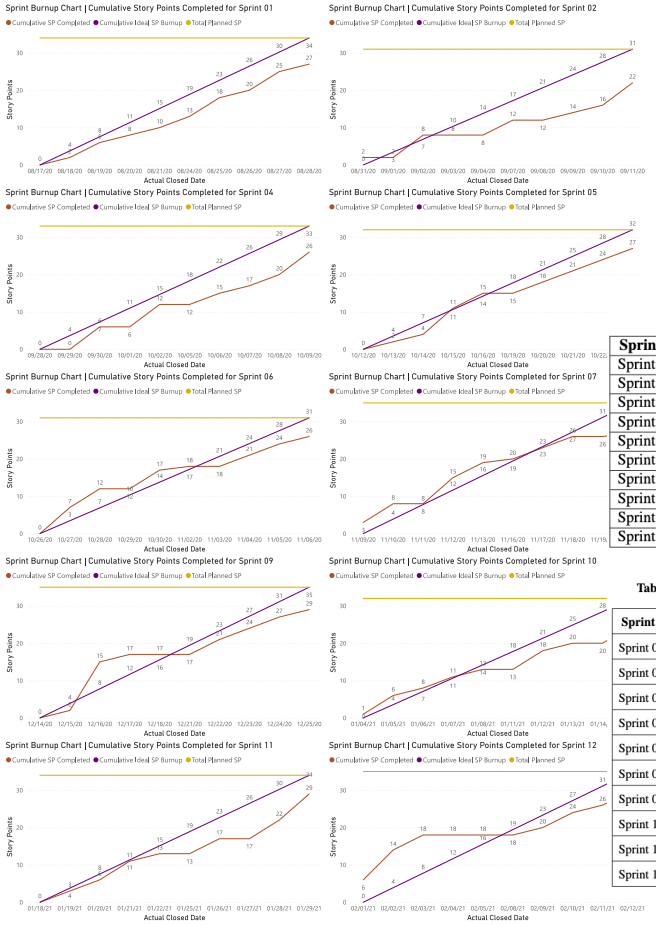


Table D.1: Monitoring CR for Scrum-driven sprints.

Sprint #	Start Date	End Date	Planned SP	Completed SP
Sprint 01	8/17/2020	8/28/2020	34	27
Sprint 02	8/31/2020	9/11/2020	31	22
Sprint 04	9/28/2020	10/9/2020	33	26
Sprint 05	10/12/2020	10/23/2020	32	27
Sprint 06	10/26/2020	11/6/2020	31	26
Sprint 07	11/9/2020	11/20/2020	35	27
Sprint 09	12/14/2020	12/25/2020	35	29
Sprint 10	01/04/2021	1/15/2021	32	25
Sprint 11	01/18/2021	1/29/2021	34	29
Sprint 12	02/01/2021	2/12/2021	35	30

Table D.3: Monitoring Defects for Scrum-driven sprints.

Sprint #	PBI	Pre-Delivery Defects	Post-Delivery Defects
Sprint 01	16	18	3
Sprint 02	15	17	3
Sprint 04	16	13	3
Sprint 05	13	13	3
Sprint 06	42	31	6
Sprint 07	20	16	3
Sprint 09	26	28	5
Sprint 10	23	24	5
Sprint 11	27	19	4
Sprint 12	41	28	5

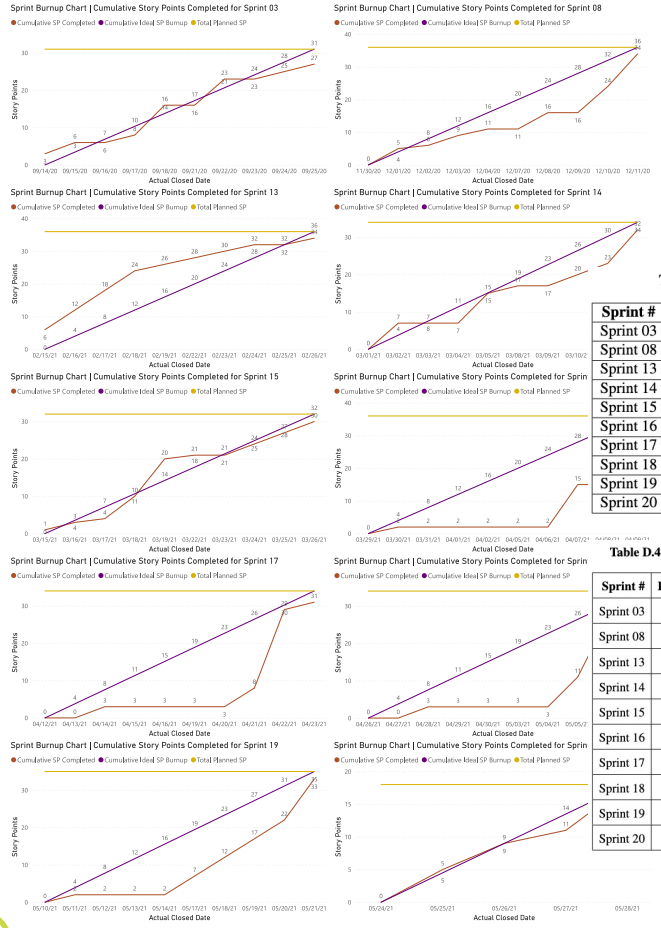


Table D.2: Monitoring CR for sMBSAP-driven sprints.

Sprint #	Start Date	End Date	Planned SP	Completed SP
Sprint 03	9/14/2020	9/25/2020	31	27
Sprint 08	11/30/2020	12/11/2020	36	34
Sprint 13	02/15/2021	2/26/2021	36	34
Sprint 14	03/1/2021	3/12/2021	34	32
Sprint 15	03/15/2021	3/26/2021	32	30
Sprint 16	3/29/2021	4/9/2021	36	33
Sprint 17	04/12/2021	4/23/2021	34	31
Sprint 18	04/26/2021	5/7/2021	34	32
Sprint 19	05/10/2021	5/21/2021	35	33
Sprint 20	05/24/2021	5/28/2021	18	18

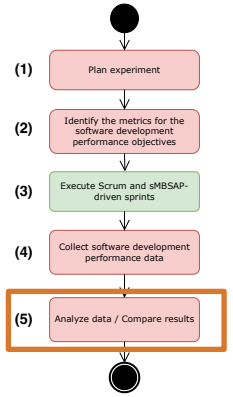
Table D.4: Monitoring Defects for sMBSAP-driven sprints.

Sprint #	PBI	Pre-Delivery Defects	Post-Delivery Defects
Sprint 03	17	6	1
Sprint 08	26	24	3
Sprint 13	32	10	2
Sprint 14	15	6	1
Sprint 15	16	14	3
Sprint 16	11	10	2
Sprint 17	17	13	2
Sprint 18	8	7	1
Sprint 19	18	12	2
Sprint 20	24	4	0

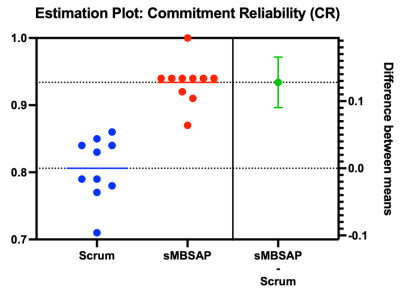
Scrum Data → Reliability of Estimation
Productivity
Defect Rate ← sMBSAP Data

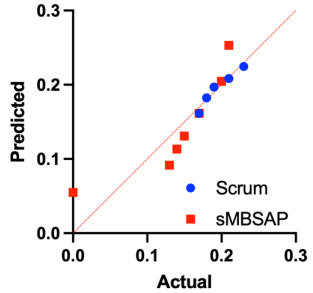
Techniques and Procedures (5 of 5)

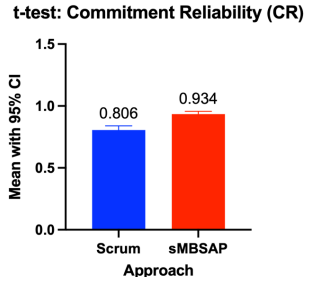
⑤ Analyze Data and Compare Results

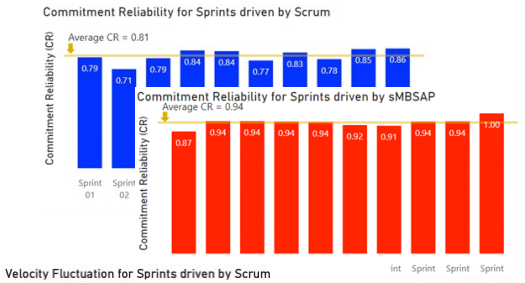


Reliability of Estimation

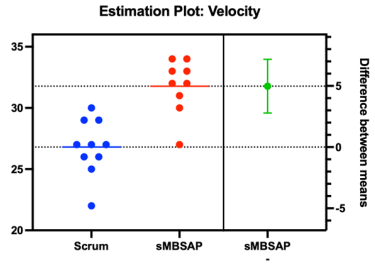


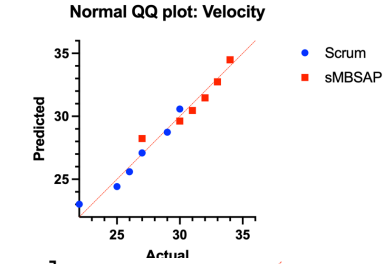


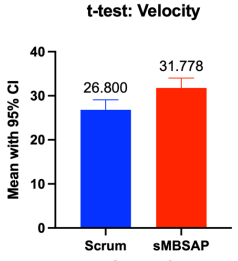




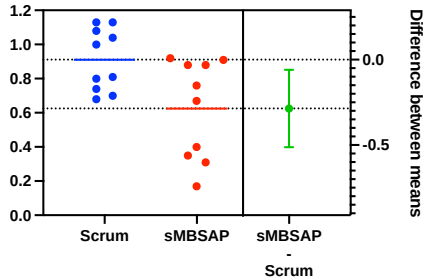
Productivity

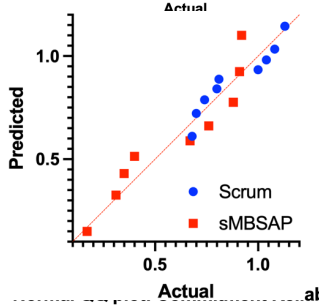


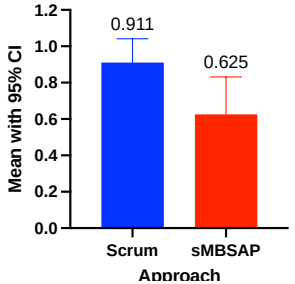


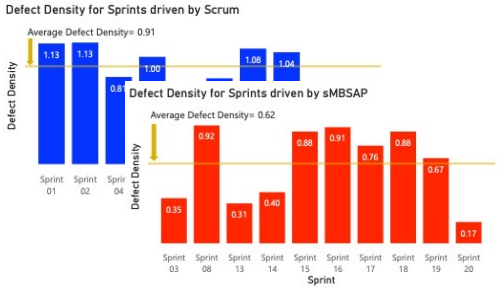


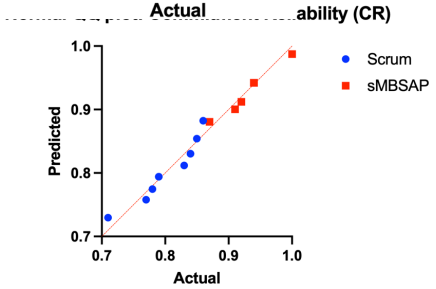
Defect Leakage

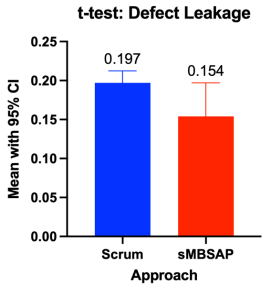


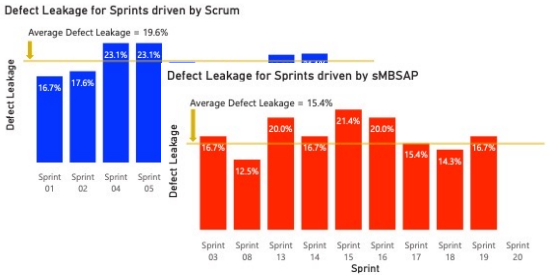












Quality of Research Design

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.

Minimizing Threat to Reliability [216]. Minimizing Threat to Validity [223,235,247]. Minimizing Threat to Statistical Conclusion Validity [234-236,242,248-253]

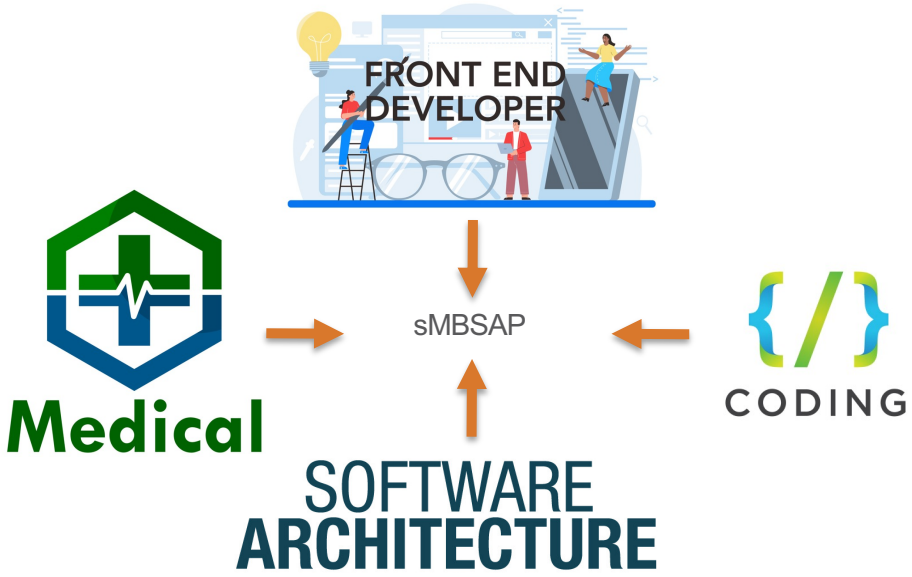
4

Research Outcome and Results

Stage 1: sMBSAP Implementation

Stage 2: Experiment Results

sMBSAP Implementation



Find a common ground of communication and collaboration between stakeholders with different backgrounds.

	Scrum	MBSAP	sMBSAP
Focus	Collaboration, Iterative and Incremental Development, Continuous Improvement, Customer-Centricity, Prioritizing Delivering Value to Customer		
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
Artifcts	Product/Sprint Backlog, and Product Increment	OV, LV, PV, and Product Increment	Product/Sprint Backlog, OV, LV, PV, and Product Increment

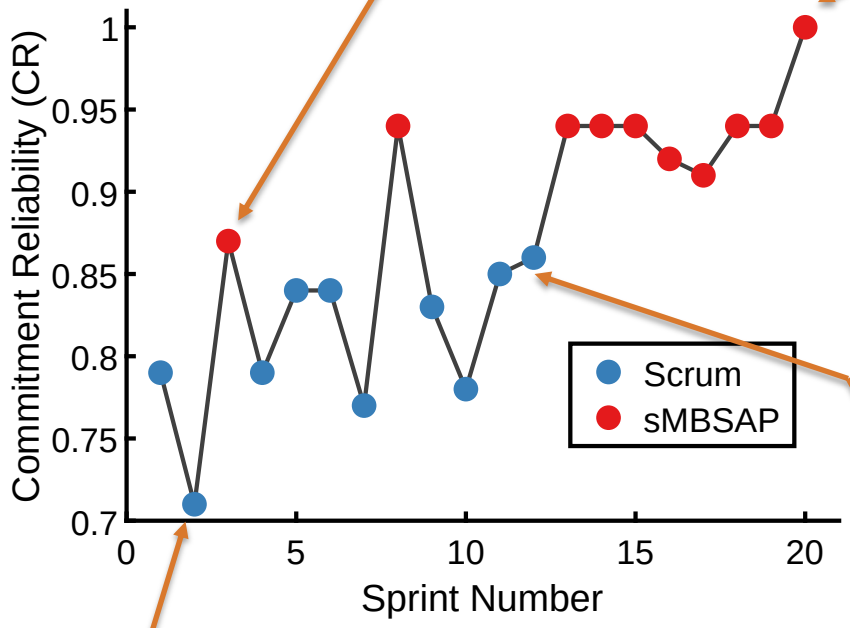
Build on and leverage the best capabilities of both methods.

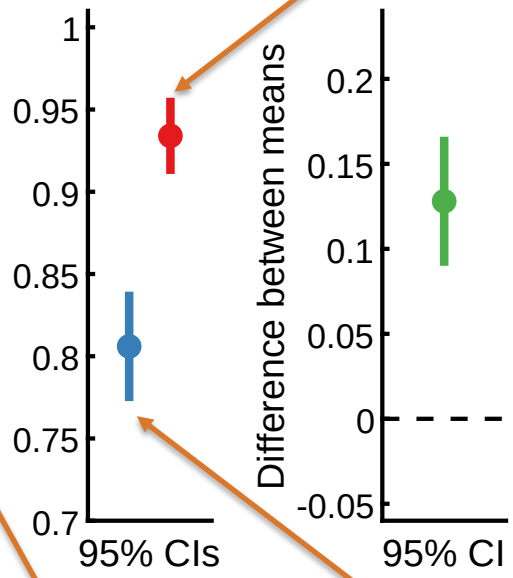
Hypothesis 1: Reliability of Estimation

lowest CR for sMBSAP=0.87

highest CR for sMBSAP=1.0

M = 0.93 (*SD* = 0.032)





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

M = 0.81 (*SD* = 0.046)

lowest CR for Scrum=0.71

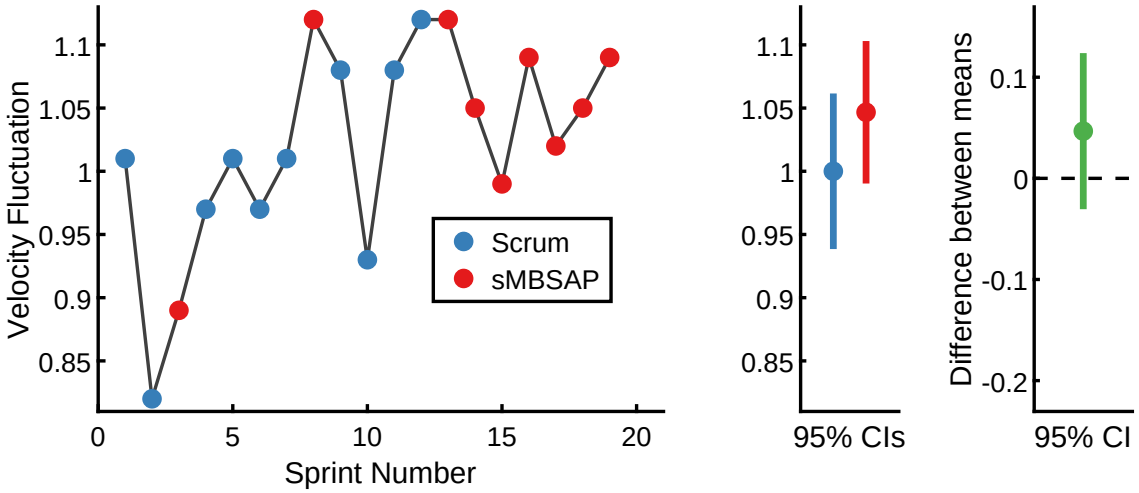
highest CR for Scrum=0.86

Hypothesis 2: Productivity

$$CLOC\ per\ hour(Scrum) = 58.03$$

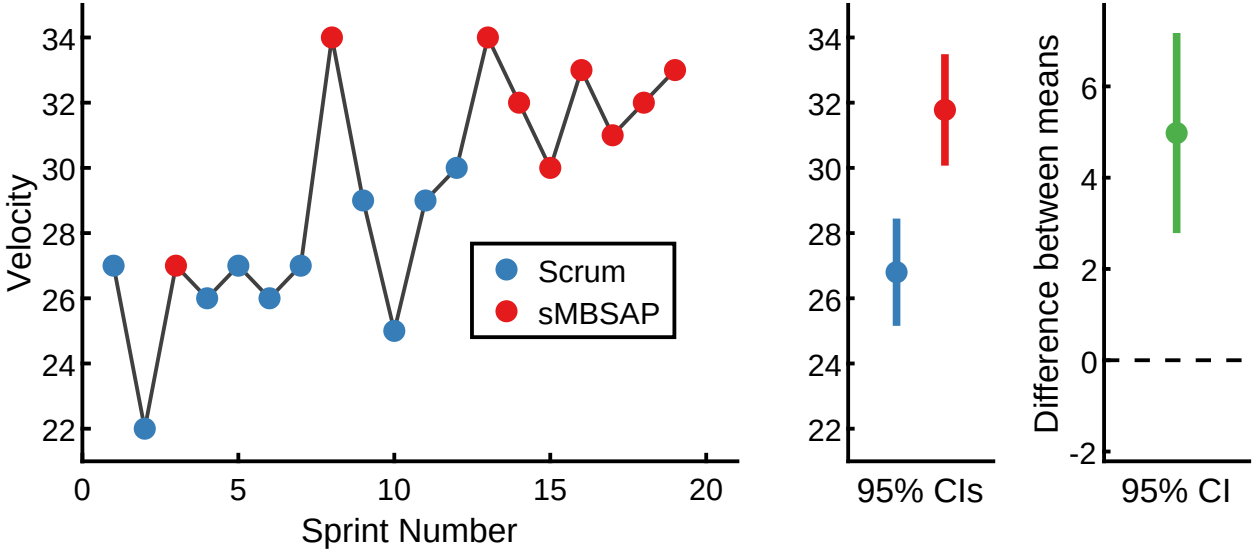
$$CLOC\ per\ hour(sMBSAP) = 71.1$$

sMBSAP-driven sprints were associated with a numerically larger *CLOC per hour* than the Scrum-driven sprints.



The results from this study suggest no noticeable difference in the *VF* between the Scrum and sMBSAP-driven sprints.

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

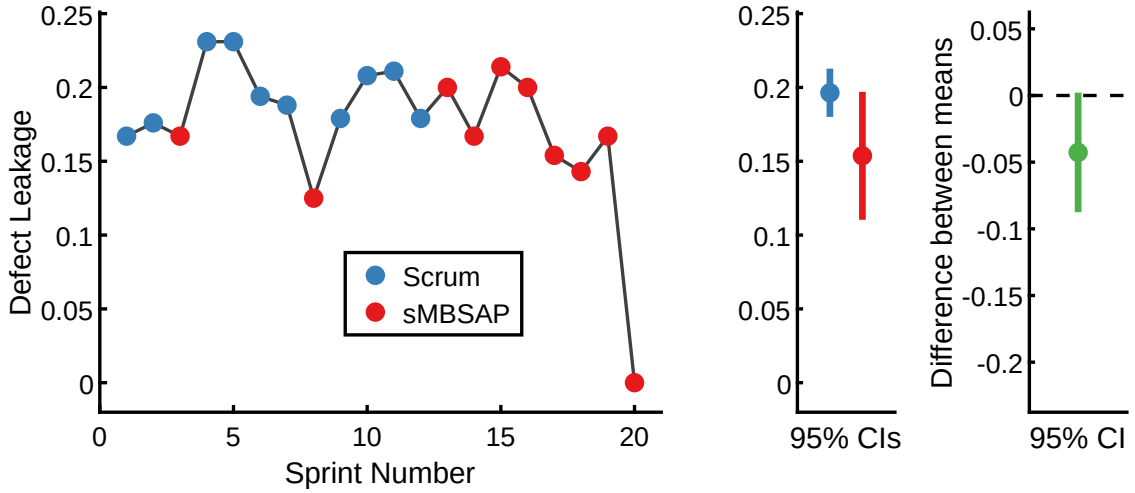


Hypothesis 3: Defect Rate

$$DD\ in\ KLOC\ (Scrum) = 4.57$$

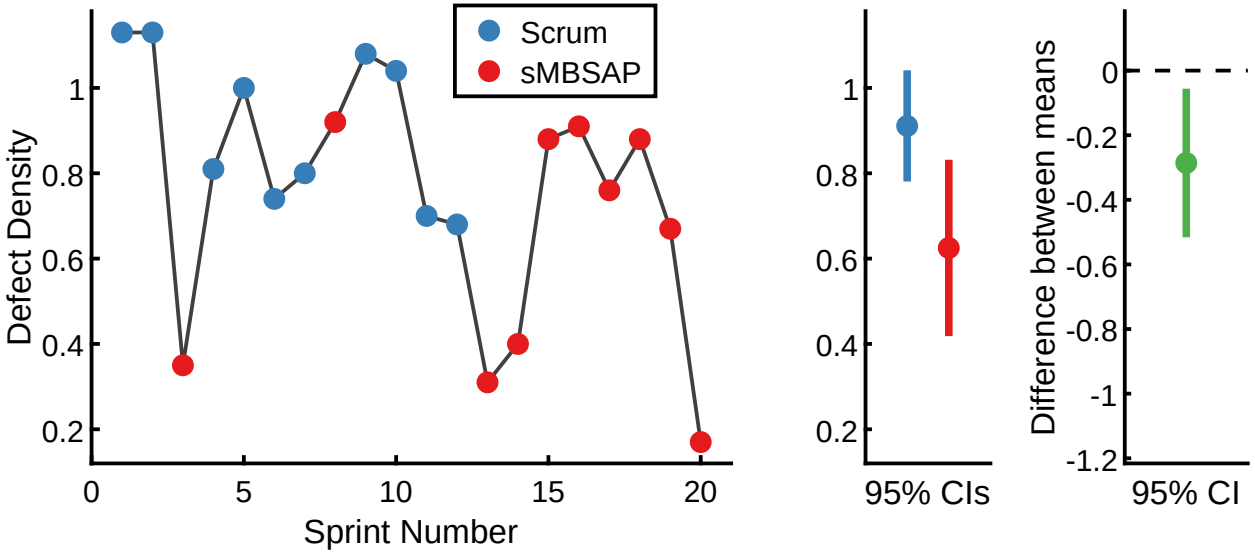
$$DD\ in\ KLOC(sMBSAP) = 2.24$$

sMBSAP-driven sprints were associated with a numerically smaller *DD* using the KLOC as size than the Scrum-driven sprints.



sMBSAP-driven sprints were associated with a statistically significant smaller mean *DL* than the Scrum-driven sprints.

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



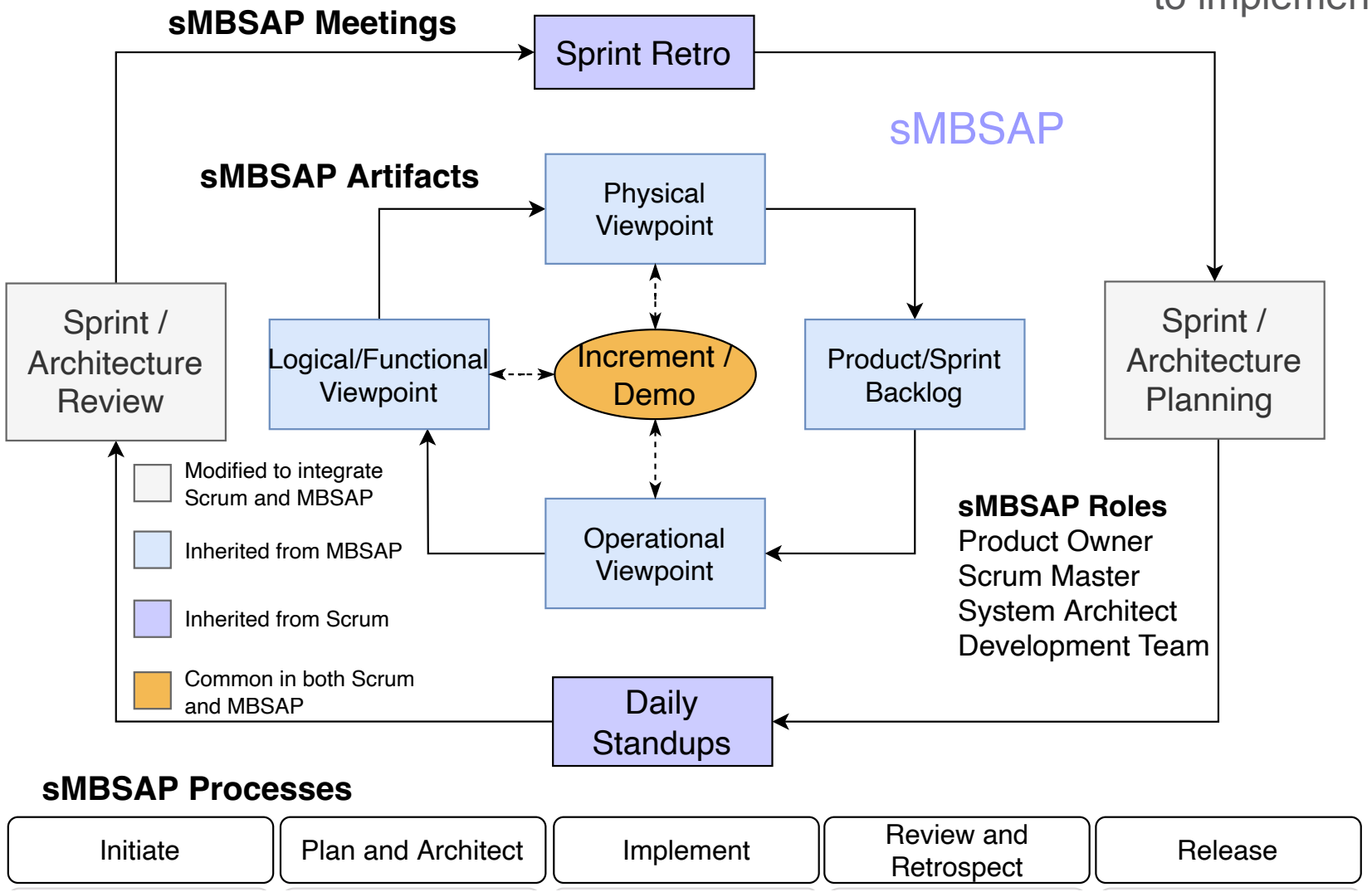
5

Conclusions

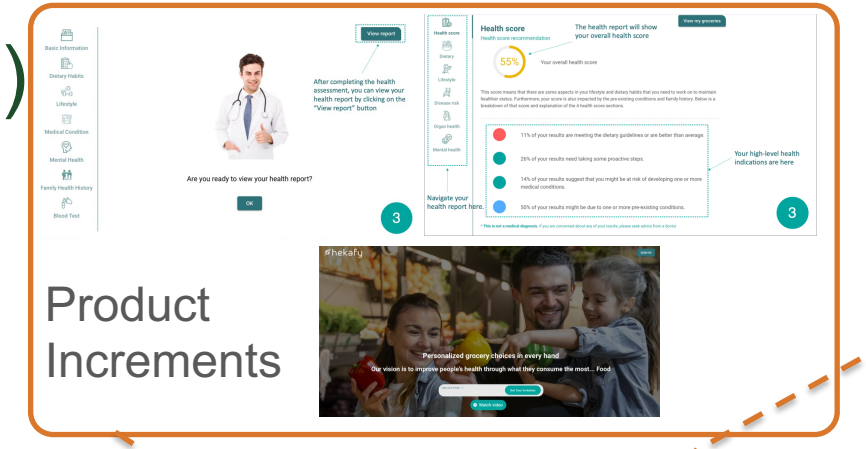
Summary, Contribution, Future Work and Publications

Summary (1 of 2)

Practical and operational method to implement Agile MBSE

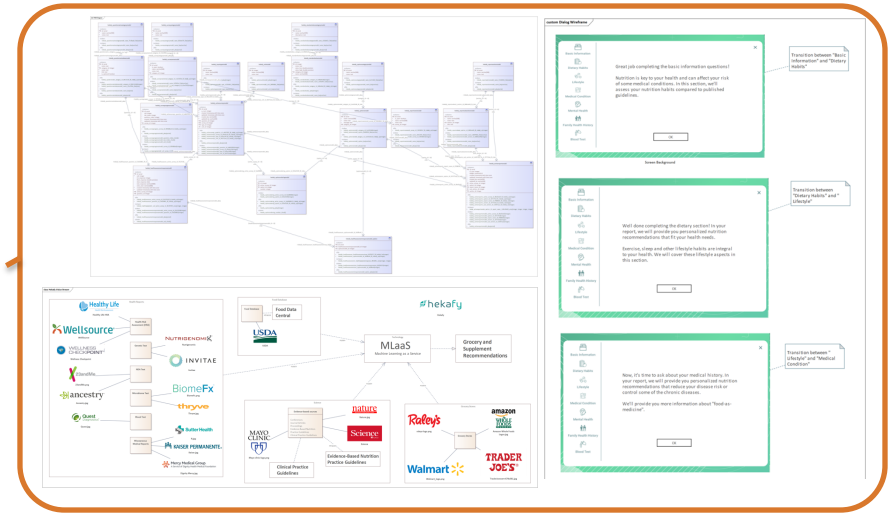


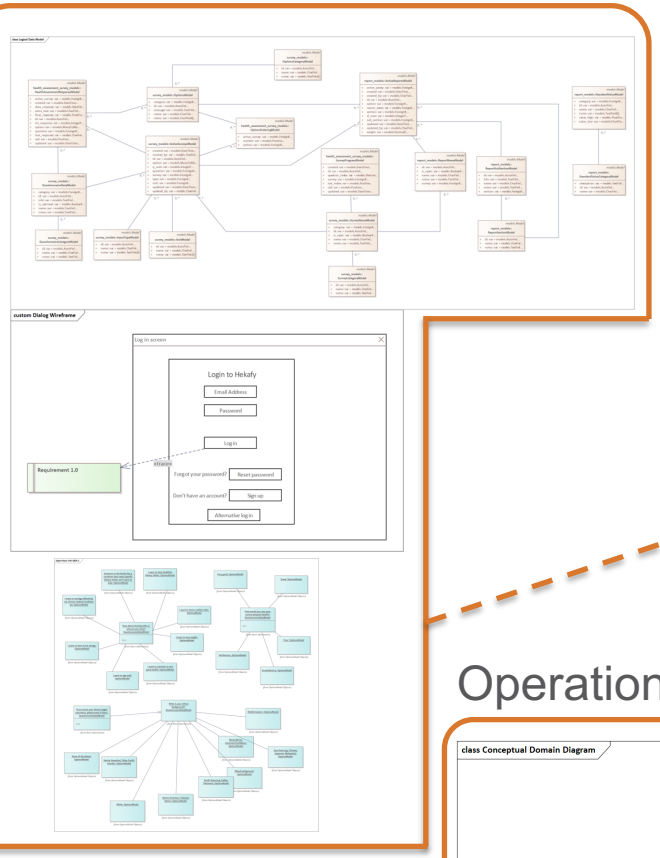
Summary (1 of 2)



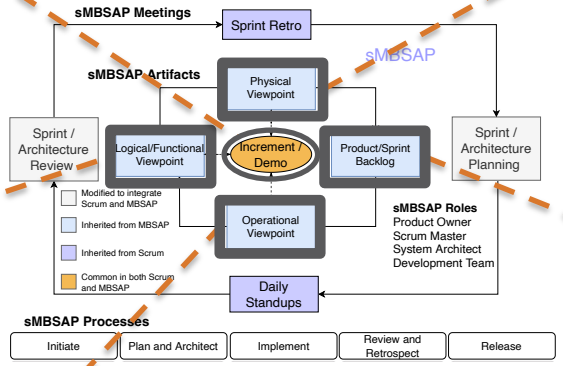
Product Increments

Physical

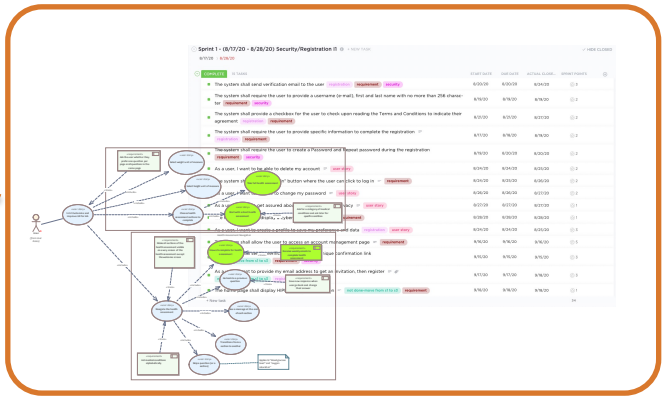




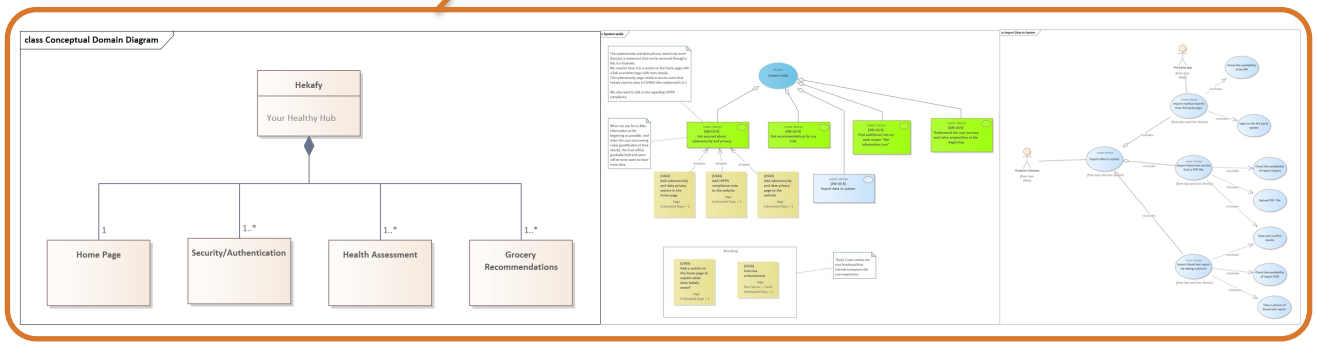
Logical



Product/Sprint Backlog



Operational

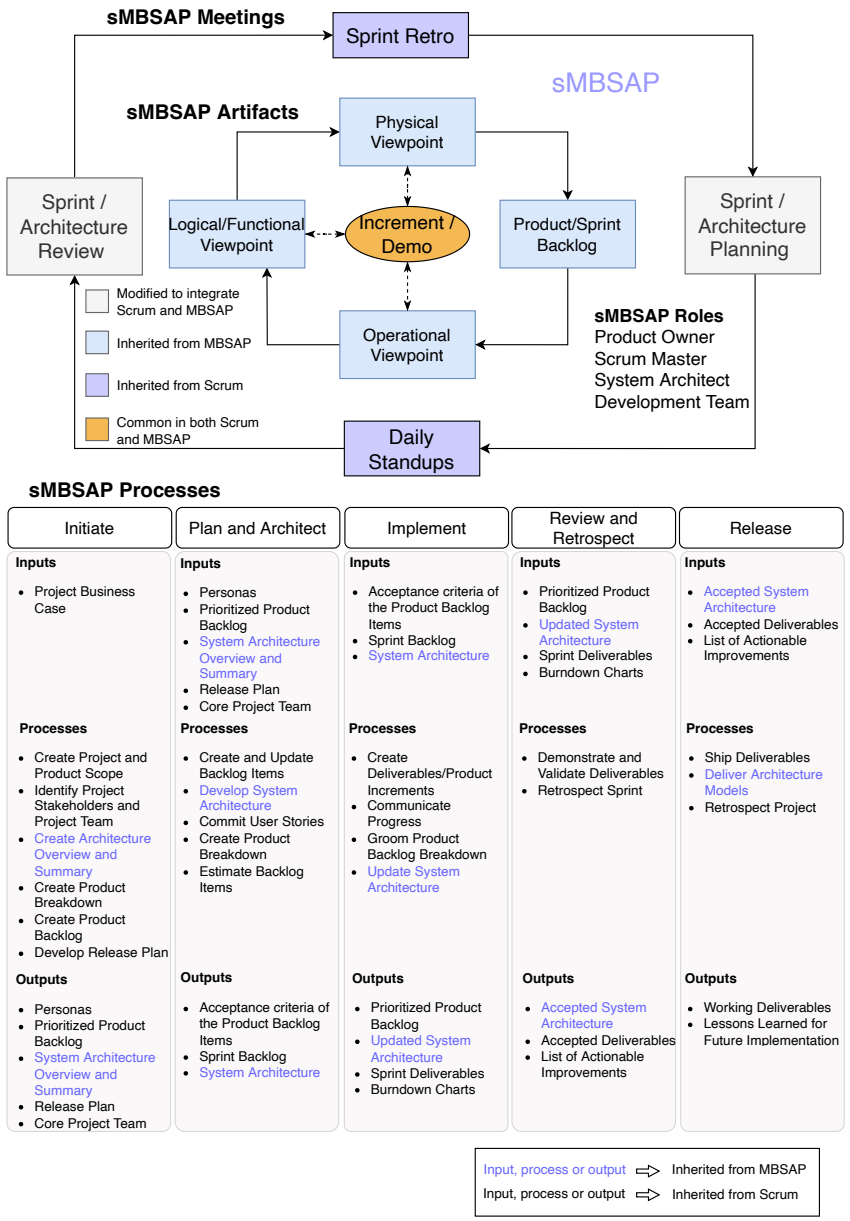


Summary (2 of 2)

Metric	Scrum	sMBSAP	Result
Commitment Reliability	0.81	0.94	sMBSAP has a statistically significant larger mean
Velocity	26.8	30.4	sMBSAP has a statistically significant larger mean
Velocity Fluctuation	0.08	0.07	No considerable difference
CLOC per Hour	58.03	71.1	sMBSAP has a numerically larger ratio
Defect Density (PBIs)	0.91	0.62	sMBSAP has a statistically significant smaller mean
Defect Density (KLOC)	4.57	2.24	sMBSAP has a numerically smaller ratio
Defect Leakage	19.6%	15.4%	sMBSAP has a statistically significant smaller mean

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.

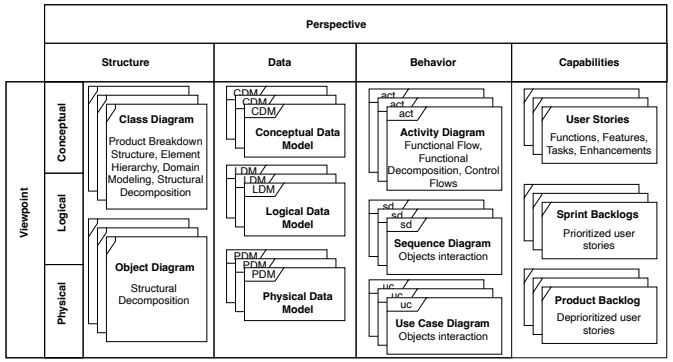


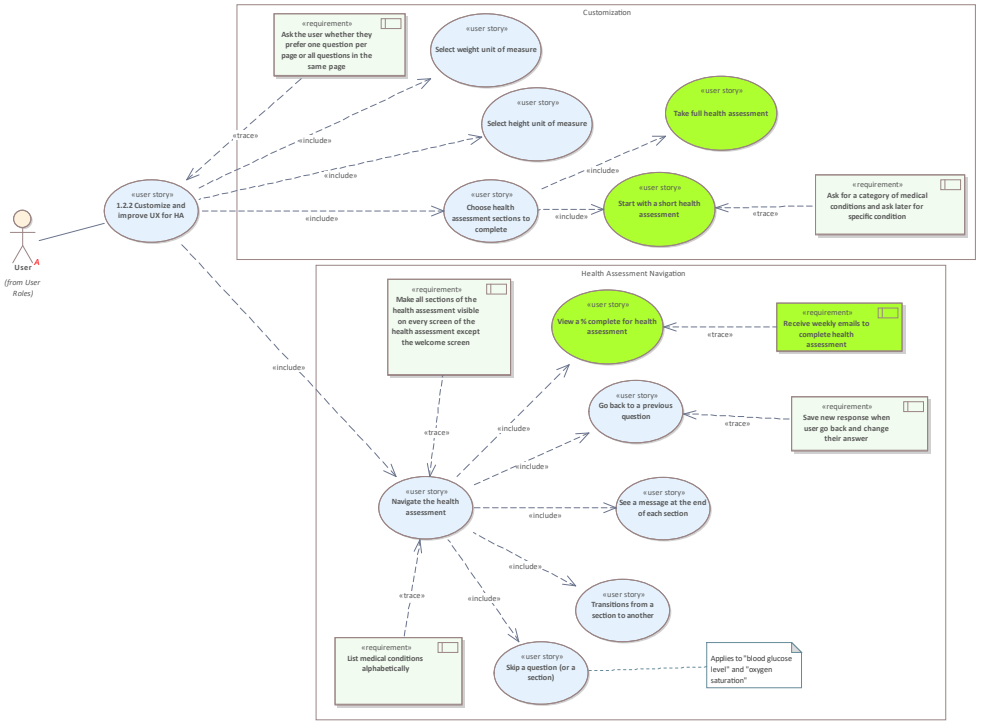
Research Contribution

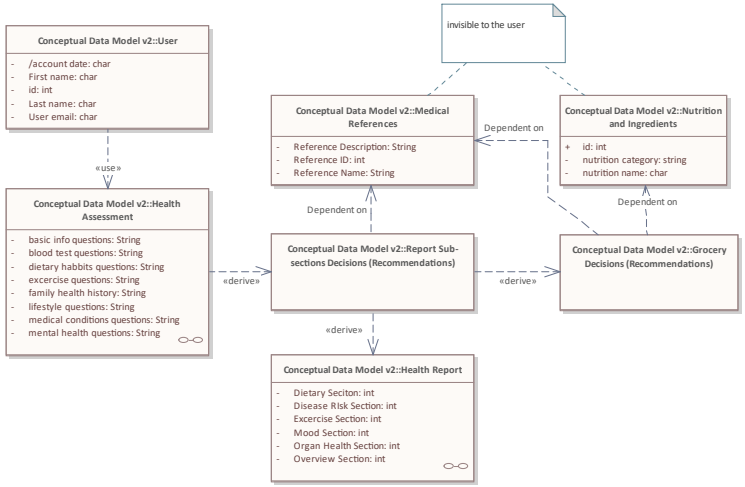
- **A practical and operational method for merging Agile and MBSE** validated through architecting and developing a prototype for a health tech system. In parallel, the results suggest that sMBSAP is a middle ground hat is more aligned with federal and state regulations, as it addresses the technical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

		Perspective			
		Structure	**Data**	**Behavior**	**Capabilities**
Viewpoint	**Operational**	**Classifications:** Hierarchy Levels **Product Breakdown:** Product Breakdown Structure, Structural Decompositions **Class Diagrams:** Domain Diagram with Specifications **Object Diagrams:** Decomposition of Domains to the level of Classes	**Classifications**: Hierarchy Levels **Conceptual Data Model (CDM):** Primary Data Entities, Foundation Classes **Logical Data Model (LDM):** Modeling Foundation Classes, Attributes and Operations **Physical Data Model (PDM):** Database Structure	**Use Case (UC) Diagram:** with Specifications **Activity Diagrams (AD):** with multiple Domains and Actors **Sequence Diagram (SD):** Modeling non-stateful behavior of a group of classes	**Classifications:** Hierarchy Levels, User Story Types **Requirements**: Requirements Hierarchy, Requirement IDs, Requirements Texts, Verification Methods, Parents, Requirements Allocations
	Functional				
	Physical				

		Perspective			
		Structure	**Data**	**Behavior**	**User Stories**
Viewpoint	**Operational**	**Product Breakdown:** Initiatives, Epics, and Stories	Data Model Slides	**User Stories** Flowcharts	**User Stories (US), Product Backlog, Sprint Backlog**
	Functional				
	Physical				

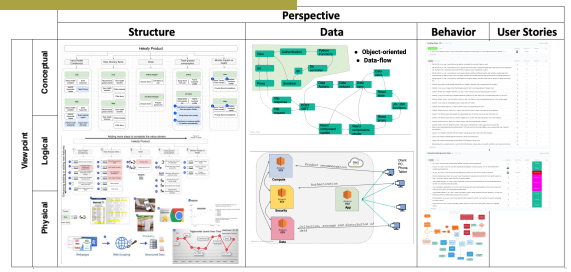


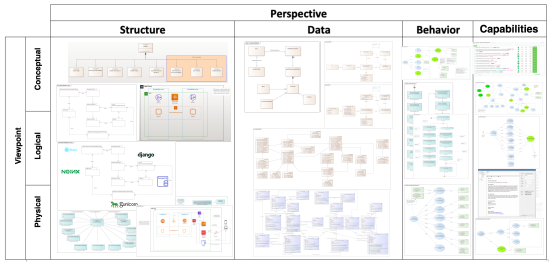


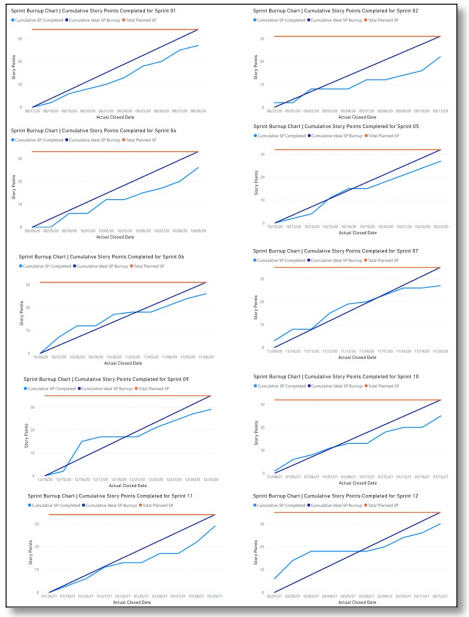


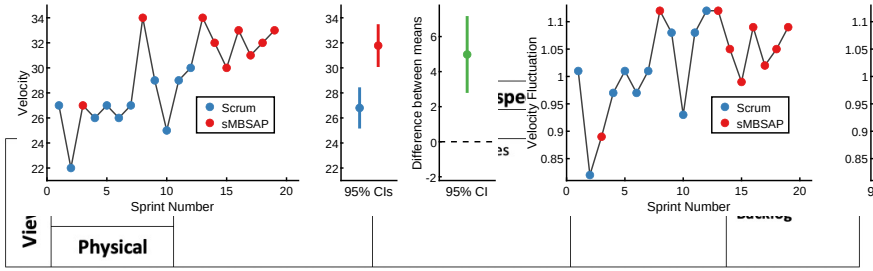
Research Contribution

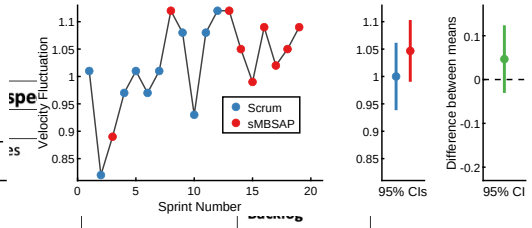
- **A practical and operational method for merging Agile and MBSE** validated through architecting and developing a prototype for a health tech system. In parallel, the results suggest that sMBSAP is a middle ground hat is more aligned with federal and state regulations, as it addresses the technical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

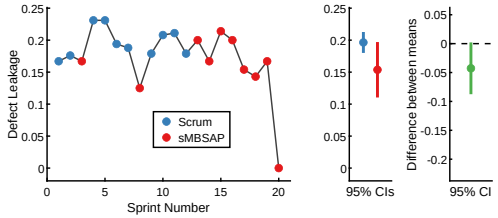


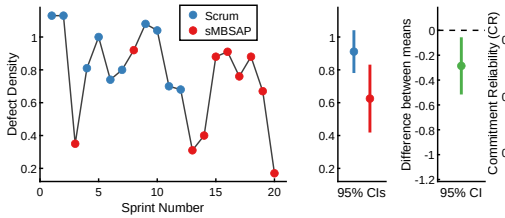


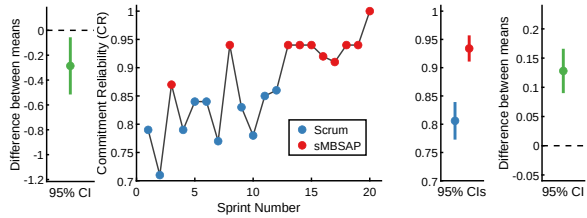










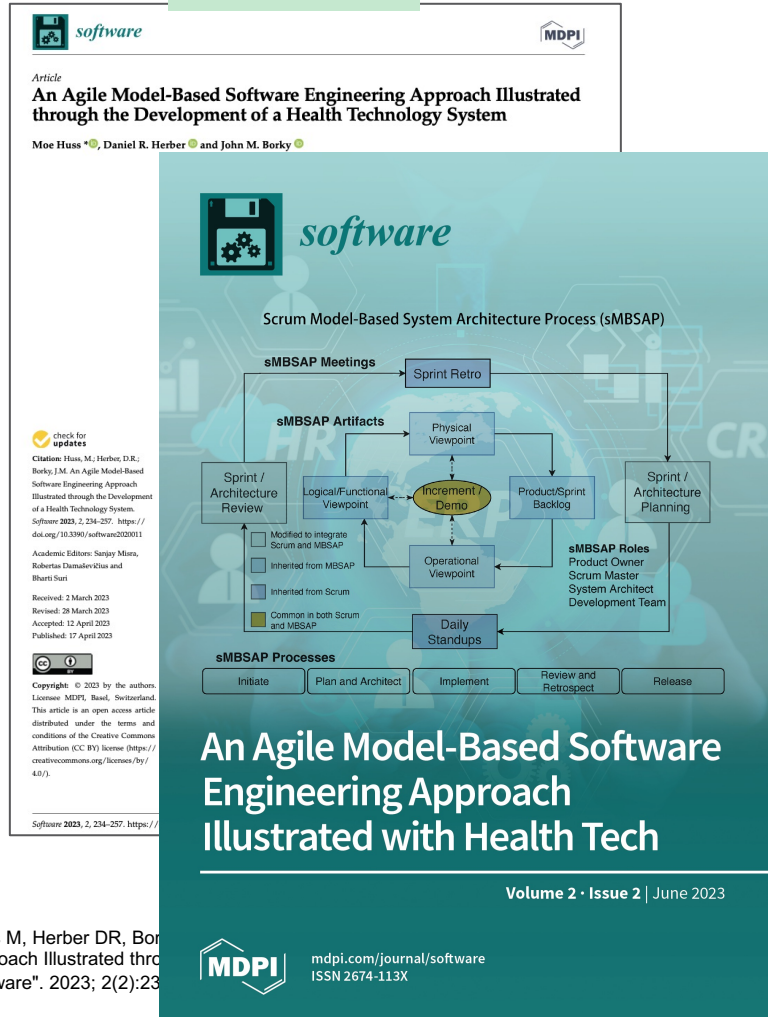


Future Work

- 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.

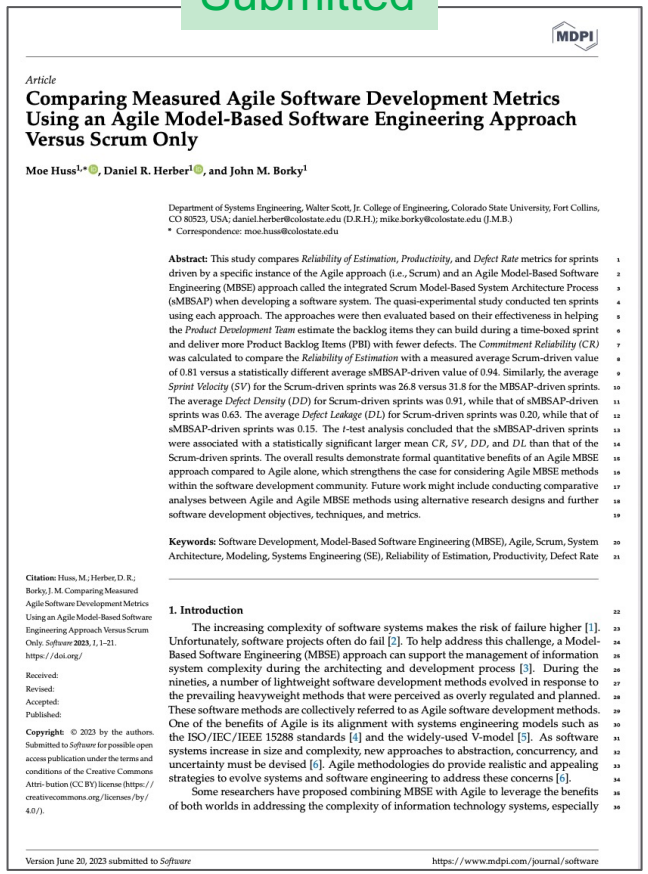
Journal Publications

Published



Huss M, Herber DR, Bor
Approach Illustrated thro
Software". 2023; 2(2):23

Submitted



Huss M, Herber DR, Borky JM. "Comparing Measured Agile Software Development Metrics Using an Agile Model-Based Software Engineering Approach Versus Scrum Only". Submitted to Software, 2023.

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)

- [1] J. M. Borky and T. H. Bradley, Effective Model-Based Systems Engineering. Springer, 2019, doi: 10.1007/978-3-319-95669-5.
- [2] M. Huss, D. R. Herber, and J. M. Borky, “An Agile model-based software engineering approach illustrated through the development of a health technology system,” Software, vol. 2, no. 2, pp. 234–257, 2023, doi: 10.3390/software2020011.
- [3] M. Huss, D. R. Herber, and J. M. Borky, “Comparing measured agile software development metrics using an agile model- based software engineering approach versus scrum only,” Software, submitted, submitted.
- [18] R. Turner, “Toward agile systems engineering processes,” Crosstalk. The Journal of Defense Software Engineering, pp. 11–15, 2007.
- [22] V. Salehi and S. Wang, “Munich Agile MBSE concept (MAGIC),” Proceedings of the De- sign Society: International Conference on Engineering Design, vol. 1, no. 1, pp. 3701–3710, Jul. 2019, doi: 10.1017/dsi.2019.377.
- [25] B. P. Douglass, Agile Model-Based Systems Engineering Cookbook, 1st ed. Packt Publish- ing, Mar. 2021.
- [37] “What Can the State Do to Prevent Future IT Project Delays and Cost Overruns,” Room 444, p. 9, 2015.
- [38] N. B. Ruparelia, “Software development lifecycle models,” ACM SIGSOFT Software Engi- neering Notes, vol. 35, no. 3, pp. 8–13, May 2010, doi: 10.1145/1764810.1764814.
- [56] K. Schwaber, “Scrum development process,” in Business object design and implementation. Springer, 1997, pp. 117–134.
- [58] U.S. Government Accountability Office, “Agile assessment guide: Best practices for agile adoption and implementation,” Washington, DC, Tech. Rep. GAO-20-590G, Sep.
- [60] D. Azhar, E. Mendes, and P. Riddle, “A systematic review of web resource estimation,” in International Conference on Predictive Models in Software Engineering, 2012, pp. 49–58, doi: 10.1145/2365324.2365332.
- [61] M. Jorgensen and M. Shepperd, “A systematic review of software development cost estima- tion studies,” IEEE Transactions on Software Engineering, vol. 33, no. 1, pp. 33–53, 2006, doi: 10.1109/TSE.2007.256943.
- [62] K. Molokken and M. Jorgensen, “A review of software surveys on software effort estima- tion,” in International Symposium on Empirical Software Engineering, 2003, pp. 223–230, doi: 10.1109/ISESE.2003.1237981.
- [63] J. Wen, S. Li, Z. Lin, Y. Hu, and C. Huang, “Systematic literature review of machine learning based software development effort estimation models,” Information and Software Technol- ogy, vol. 54, no. 1, pp. 41–59, 2012, doi: 10.1016/j.infsof.2011.09.002.
- [64] M. Cohn, Agile Estimating and Planning. Pearson Education, 2005.
- [74] M. Tiwari, “Understanding Scrum metrics and KPIs,” Online, Nov. 2019, medium. url: https://medium.com/@techiemanoj4/understanding-scrum-metrics-and-kpis- fcf56833d488.
- [75] J. Smart, “To transform to have agility, dont do a capital A, capital T Agile transformation,” IEEE Software, vol. 35, no. 6, pp. 56–60, 2018, doi: 10.1109/MS.2018.4321245.
- [76] M. Khanzadi, M. M. Shahbazi, and H. Taghaddos, “Forecasting schedule reliability using the reliability of agents’ promises,” Asian Journal of Civil Engineering, vol. 19, pp. 949– 962, 2018, doi: 10.1007/s42107-018-0075-7.

References (2 of 3)

- [77] C. Chen, S. Housley, P. Sprague, and P. Goodlad, “Introducing lean into the UK Highways Agency’s supply chain,” Proceedings of the Institution of Civil Engineers - Civil Engineering, vol. 165, no. 5, pp. 34–39, 2012, doi: 10.1680/cien.11.00013.
- [99] S. Wagner and M. Ruhe, “A systematic review of productivity factors in software develop- ment,” arXiv, 2018, doi: 10.48550/arXiv.1801.06475.
- [103] H. Huijgens and R. v. Solingen, “A replicated study on correlating agile team velocity mea- sured in function and story points,” in International Workshop on Emerging Trends in Software Metrics, 2014, pp. 30–36, doi: 10.1145/2593868.2593874.
- [104] G. B. Alleman, M. Henderson, and R. Seggelke, “Making Agile development work in a government contracting environment-measuring velocity with earned value,” in Agile Development Conference, Jun. 2003, pp. 114–119, doi: 10.1109/ADC.2003.1231460.
- [105] T. Javdani, H. Zulzalil, A. A. A. Ghani, A. B. M. Sultan, and R. M. Parizi, “On the current measurement practices in agile software development,” arXiv, 2013, doi: 10.48550/arXiv.1301.5964.
- [109] F. Albero Pomar, J. A. Calvo-Manzano, E. Caballero, and M. Arcilla-Cobián, “Under- standing sprint velocity fluctuations for improved project plans with scrum: a case study,” Journal of Software: Evolution and Process, vol. 26, no. 9, pp. 776–783, 2014, doi: 10.1002/smr.1661.
- [110] “IEEE standard glossary of software engineering terminology,” IEEE, Standard, 1990, doi: 10.1109/IEEESTD.1990.101064.
- [111] P. Tripathy and K. Naik, Software Testing and Quality Assurance: Theory and Practice. John Wiley & Sons, 2011, doi: 10.1002/9780470382844.
- [112] M. McDonald, R. Musson, and R. Smith, The Practical Guide to Defect Prevention. Mi- crosoft Press, 2007.
- [116] P. Mohagheghi, R. Conradi, O. M. Killi, and H. Schwarz, “An empirical study of software reuse vs. defect-density and stability,” in International Conference on Software Engineering, 2004, pp. 282–291, doi: 10.1109/ICSE.2004.1317450.
- [117] N. Nagappan and T. Ball, “Use of relative code churn measures to predict system defect density,” in International Conference on Software Engineering, 2005, pp. 284–292, doi: 10.1145/1062455.1062514.
- [118] S. M. A. Shah, M. Morisio, and M. Torchiano, “An overview of software defect density: a scoping study,” in Asia-Pacific Software Engineering Conference, vol. 1, 2012, pp. 406–415, doi: 10.1109/APSEC.2012.93.
- [120] T.M.FehlmannandE.Kranich,“Measuringdefectsinasixsigmacontext,”inInternational Conference on Lean Six Sigma, 2014.
- [122] J. S. Collofello, Z. Yang, J. D. Tvedt, D. Merrill, and I. Rus, “Modeling software testing processes,” in IEEE Fifteenth Annual International Phoenix Conference on Computers and Communications, 1996, pp. 289–293, doi: 10.1109/PCCC.1996.493647.
- [123] P. M. Jacob and M. Prasanna, “A comparative analysis on black box testing strategies,” in International Conference on Information Science, 2016, pp. 1–6, doi: 10.1109/INFOSCI.2016.7845290.

References (3 of 3)

- [124] M. S. Rawat and S. K. Dubey, “Software defect prediction models for quality improvement: a literature study,” International Journal of Computer Science Issues, vol. 9, no. 5, pp. 288– 296, 2012.
- [125] V. Vashisht, M. Lal, and G. S. Sureshchandar, “A framework for software defect prediction using neural networks,” Journal of Software Engineering and Applications, vol. 8, no. 8, pp. 384–394, 2015, doi: 10.4236/jsea.2015.88038.
- [151] R.V.AnandandM.Dinakaran,“IssuesinscrumAgiledevelopmentprinciplesandpractices in software development,” Indian Journal of Science and Technology, vol. 8, no. 35, pp. 1–5, 2015, doi: 10.17485/ijst/2015/v8i35/79037.
- [152] Digital.ai, Feb 2022, url: https://digital.ai/resource-center/analyst-reports/state-of-agile- report/.
- [153] M. Pittman, “Lessons learned in managing object-oriented development,” IEEE Software, vol. 10, no. 1, pp. 43–53, 1993.
- [154] G. Booch, Object solutions: managing the object-oriented project. Addison Wesley Long- man Publishing Co., Inc., 1995.
- [155] I. S. Graham and A. Graham, Migrating to object technology. Addison-Wesley Longman Publishing Co., Inc., 1995.
- [156] H. Takeuchi and I. Nonaka, “The new new product development game,” Harvard business review, vol. 64, no. 1, pp. 137–146, 1986.
- [203] B. Selic, “Agile documentation, anyone?” IEEE Software, vol. 26, no. 6, pp. 11–12, 2009, doi: 10.1109/MS.2009.167.
- [216] M. Saunders, P. Lewis, and A. Thornhill, Research Methods, 8th ed. Pearson Education, 2019.
- [222] J. Brannen, “Combining qualitative and quantitative approaches: an overview,” Mixing methods: Qualitative and quantitative research, pp. 3–37, 2017.
- [231] H. Tang, Engineering research: Design, methods, and publication. John Wiley & Sons, 2020.
- [234] S. E. Maxwell, H. D. Delaney, and K. Kelley, Designing experiments and analyzing data: A model comparison perspective. Routledge, 2017.
- [235] D. T. Campbell and J. C. Stanley, Experimental and Quasi-Experimental Designs for Research. Houghton Mifflin Company, 1963.
- [236] C. S. Reichardt, Quasi-Experimentation: A Guide to Design and Analysis. Guilford Publications, 2019.
- [237] D. Schwartz, B. Fischhoff, T. Krishnamurti, and F. Sowell, “The hawthorne effect and en- ergy awareness,” Proceedings of the National Academy of Sciences, vol. 110, no. 38, pp. 15 242–15 246, 2013, doi: 10.1073/pnas.1301687110.
- [242] A. P. Rovai, J. D. Baker, and M. K. Ponton, Social Science Research Design and Statistics. Watertree Press, 2013.
- [247] E. C. Matthay and M. M. Glymour, “A graphical catalog of threats to validity: Linking social science with epidemiology,” Epidemiology, vol. 31, no. 3, pp. 376–384, 2020, doi: 10.1097/EDE.0000000000001161.
- [248] D. C. Howell, “Chi-square test: analysis of contingency tables,” in International Encyclo- pedia of Statistical Science. Springer, 2011, pp. 250–252, doi: 10.1007/978-3-642-04898-2_174.
- [253] J.L.Gastwirth,Y.R.Gel,andW.Miao,“TheimpactofLevene’stestofequalityofvariances on statistical theory and practice,” Statistical Science, vol. 24, no. 3, pp. 343–360, 2009, doi: 10.1214/09-STS301.

Thank you

Colorado State University