Maximizing Value Creation in Agile Sprints

by

Nithin Thekkupadam Narayanan
B. Tech. Mechanical Engineering, College of Engineering, Thiruvananthapuram

Submitted to the System Design & Management Program
In Partial Fulfillment of the Requirements for the Degree of

Master of Science in Engineering and Management
at the
Massachusetts Institute of Technology

February 2021

Signature of the author…………………………………………………………………………….
Nithin Thekkupadam Narayanan
System Design and Management Jan 29, 2021

Certified by………..……………………………………………………………………………….
Steven D. Eppinger
Professor of Management Science and Engineering Systems
Thesis Supervisor

Accepted by…………..……………………………………………………………………………

Joan S. Rubin
Senior Lecturer and Executive Director
MIT System Design and Management Program

Maximizing Value Creation in Agile Sprints

by

Nithin Thekkupadam Narayanan
B. Tech. Mechanical Engineering, College of Engineering, Thiruvananthapuram

Submitted to the System Design & Management Program on January 29, 2021
in partial fulfillment of the requirements for the Degree of
Master of Science in Engineering and Management

Abstract

Agile software development principles prioritize the delivery of value through working software. Earlier value creation is preferred to reduce the time to market and get sooner feedback from customers. A challenge in planning agile sprints to achieve value upfront is the tension that exists between the value, resources, size of each feature or story deliverable, and the dependencies among them. While the role of effort and resource constraints in value creation has been studied extensively, the role of dependencies has not been fully addressed in the agile context. In this thesis, we propose a framework to improve value delivery in agile software development by decoupling cyclic dependencies to achieve more robust multi-sprint plans in a scaled agile environment. We analyze this novel approach using an arbitrary test dataset to demonstrate how different decoupling methods yield different value trajectories. We also suggest an optimization method to maximize such value creation through sequencing by simultaneously considering timing, dependencies, and resource allocation. We perform a brute-force optimization approach on the test dataset to demonstrate how more rapid value creation can be achieved over multiple sprints.

Thesis Supervisor: Steven D. Eppinger

Title: Professor of Management Science and Engineering System, Sloan School of Management

Acknowledgement

This thesis would not have been possible without the support, guidance and help from a number of people.

I would first like to thank my thesis advisor, Professor Steven D. Eppinger, for giving me an opportunity to assist him in the research that led to the conception of this thesis. Over the past one year, Prof. Eppinger has provided constant support and guidance and even during the holidays which instilled confidence in me in writing this thesis. Your invaluable feedback pushed me to sharpen my thinking and brought my work to a higher level.

I also thank Professor Nitin R. Joglekar of Boston University who has been a constant source of motivation throughout my research. Along with Professor Eppinger, Professor Joglekar has guided and supported me in every step of the way. His suggestions and feedbacks helped me conceptualize key aspects of the thesis. Thank you for the inordinate number of hours you spent during the holidays to help me make this happen.

I would also like to thank Swisscom AG for enabling our research. In particular, I would like to thank Pradeep R. Fernando, Vice President of Product Management, Data, Analytics & AI, for being a wonderful collaborator and for providing key insignts that helped us frame the research problem. Your advice and feedbacks has been helpful in more ways than one.

Thank Aravind Asokan for being a wonderful friend and mentor. He has been my unofficial MIT advisor. Being an alum himself, he knew the hardships a student faces during the program and selflessly helped me throughout my MIT journey to ensure that I have a memorable experience at MIT. For which I will be forever indebted.

My fellow SDM students, Ricardo Hopker, Aparajithan Sampath, and Bruce Hetch have helped me in our attempt to implement the framework proposed in this thesis, using Matlab. Though the work remains in progress and is not a part of thesis, their help has been instrumental in giving us confidence that the implementation is possible.

But none of this would have been possible without my everpresent support system – my wife. Thank you Rimple for being a constant source of positivity and inspiration.

Table of Contents

1. Introduction

1.1 Agile Software Development

Traditional, plan-based, or "Waterfall" style engineering project management methods involve detailed up-front planning and design, which is often referred to as Big Design Up Front or BDUF. This style of project management works fine for many types of projects, such as construction and VLSI design, in which features and scope can be well defined at the start. However, for other kinds of projects, such as software development, where work involves a level of uncertainty and high change rates, traditional methods fall short. In such an environment where change is frequent, and the value that is delivered can be improved based on information that comes with these changes, Agile Project Management comes into play [1]. Beck et al. formalized the agile principles in the Agile manifesto [2], forming a basis for Agile software development.

1.2 Scaled Agile

However, as software systems and organizations have scaled, there has been a growing need to provide greater coordination ability among self-organizing development teams. Turk and France have identified several challenges and limitations that arise from the assumptions underlying agile processes [3]. These assumptions do not accurately represent a large-scale system development environment. This need has given rise to "Large-Scale Agile" methodologies, which attempt to adapt the principles of agile development to the needs to coordination in a complex, scaled enterprise environment [4].

According to the 2019 Survey on Agility [5], 70% of the respondents have indicated their ambition to integrate both Business and IT-enabled Agile transformation in the next three years. The survey received responses from more than 120 participants from 17 countries. These results show an increasing interest from organizations to adopt Agile at Scale.

There are several Scaled Agile frameworks that organizations adopt to enable their Agile transformation such as Scaled Agile Framework (SAFe) [6], Solutions for Agile Governance (Sage) [7], Scrum @ Scale [8], Spotify [9], Large Scale Scrum (LeSS) [10], and Disciplined Agile Delivery (DAD) [11]. SAFe is the most common framework, according to the Survey on Agility [5]. This research is based on the scaled agile practices at Swisscom, where SAFe is the framework that is being used.

The framework of choice for this thesis is SAFe. SAFe was introduced by Dean Leffingwell in 2011. In SAFe, programs (called Agile Release Trains, or ARTs) develop and deploy together during a Program Increment (PI), which consists of multiple (usually five 2-week) sprints [6]. The synchronicity of PIs is an essential aspect of SAFe ("develop on cadence"), and it enables teams to integrate the system on a regular basis [12].

1.3 Field study at Swisscom

The research presented in this thesis was motivated by and performed in collaboration with Swisscom. Swisscom is the leading Information, and Communications Technology and telecommunications provider in Switzerland Swisscom holds a market share of 59% for mobile, 53% for broadband internet, and 36% for TV telecommunication in its domestic residential and commercial markets [13]. Swisscom is known for its premium quality offerings.

We studied the Program Increment (PI) Planning process for the Agile Release Train (ART) “Data Lake”, which is a part of the Large Solution “Data, Analytics & AI” (DNA). Swisscom’s Data Lake is a large, centralized data repository for structured and unstructured data from a variety of source systems across Swisscom. It also provides storage, computation, and access infrastructure and services to leverage this data. The DNA large solution contains five other ARTs (four analytics-focused ARTs and one ART focused on developing applications for business users using data and AI to differentiate their market offerings), which depend on the Data Lake ART for this infrastructure and use it to develop their own products and services for Swisscom business customers. All the ARTs in the DNA Large Solution do their PI Planning together in a single event.

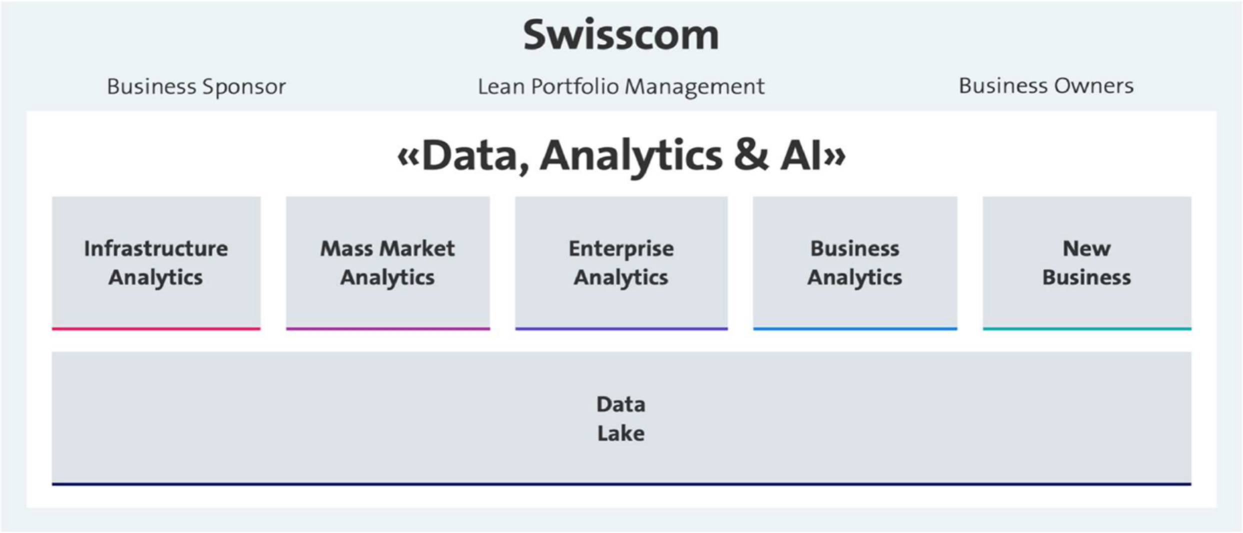


Figure 1: Data, Analytics and AI Large Solution high-level structure (source: Swisscom)

Bajpai represented the dependencies between the Data lake and the other ARTs as below [14]:

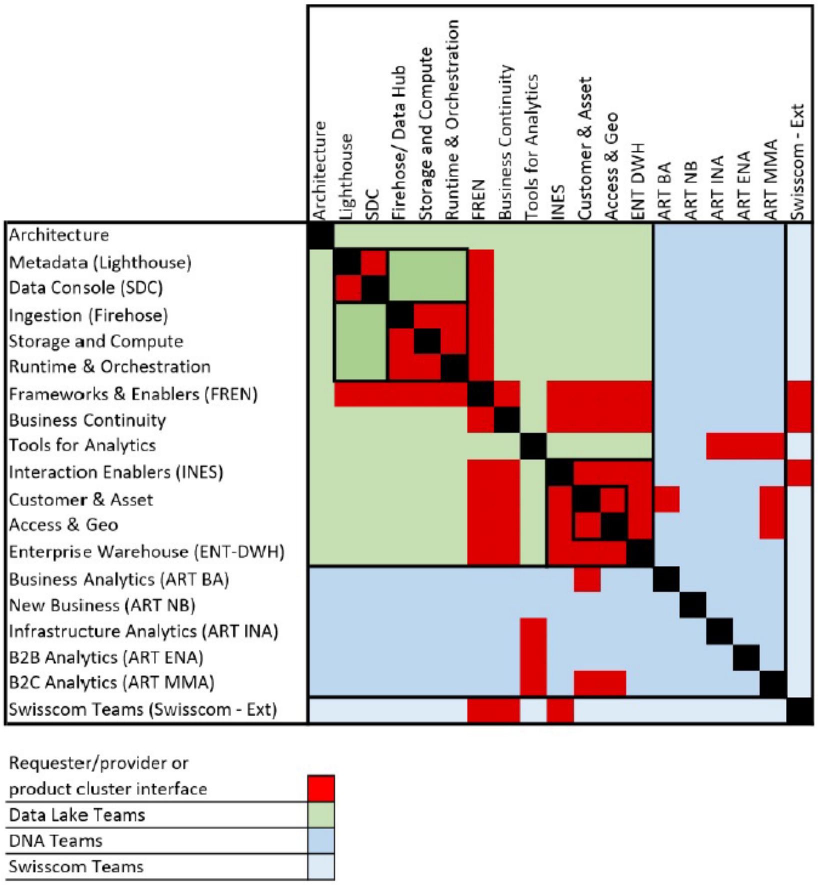


Figure 2: Nominal Interface matrix for Data Lake development teams + customer teams (source: [14])

The above diagram, Figure 2, indicates the pattern of dependencies that exists between various teams within DNA

Our discussions with the Vice President of Product Management for Data Lake program (referred to as Manager hereinafter) helped us understand various challenges that Swisscom faces during PI planning and execution. The primary concern raised was that though resource utilization is high from the beginning of a PI execution, the features are completed only towards the end of the PI. This occurs partly due to a lack of proper understanding of dependencies prior to the execution of user stories (referred to as story hereinafter). Bajpai [14] identified that there could be seven times more interactions that may be present than are represented in the PI board. Though the teams are aware of dependencies, the dependencies may not be formally registered, leading to communication gaps and coordination overhead. These gaps and overhead cause delays in the delivery of certain stories leading to an overall delay in completion of a feature.

Manager stated that developers do not like such cyclic dependencies, and it becomes an overhead for Product Owners and Scrum Masters to manage these dependencies during the sprints.

In this thesis, we propose a framework that consists of a sequencing strategy with six methods to reduce the tension created by dependencies and story size on value. We call them “Sequencing Strategies”. Using these strategies, we reduce the risk to product development by eliminating cyclic dependencies, which helps teams achieve better flexibility during the PI planning and execution. We perform an analysis of these strategies using an arbitrary test dataset to demonstrate how different strategies yield different results and perform a simple statistical analysis of these results. We also suggest an optimization method to maximize value creation by simultaneously managing effort, timing, dependencies, and resource allocation. We perform a brute-force approach on the test dataset to demonstrate how resource allocation can be achieved.

2. Literature review

2.1 Managing Value, Story Size and Resource Tension

[6] defines value as the Cost of Delay, which is the money that would be lost by delaying or not doing a job. The value of a job and its size help us identify the priority of a job using the Weighted Shortest Job First (WSJF) framework [15]. In Agile Software Development, these jobs are either features or user stories. [16] highlights the importance of value in lean thinking by stating that a job that does not provide value to customers is considered a waste.

Torrecilla-Salinas et al., in their research that proposes a new framework to estimate and plan Agile web development using a value-based approach, suggest that there exists a cost-benefit relationship between size of a story and value [17]. Hence, there is a tension between story size and the value. Wake, who defined INVEST, a popular framework used to assess the quality of a user story, highlights the importance of making the size of a user story small so as to fit within a sprint [18].

Dependencies are a major source of risk to cost-effective project execution [19]. Dependencies can lead to conflict between teams and, if left unresolved, may add unpredictability and risk to the project [20], thereby impacting the value created. Though it is desirable to eliminate all dependencies, it may not also be achievable [18]. Therefore, it is important to manage dependencies effectively.

DSM has been long considered an effective tool in managing these dependencies. According to Eppinger and Browning, DSMs can help create conciseness, enable better visualization, provide intuitive understanding, and enable better analysis and flexibility [21]. In their book, they have showcased how DSM can help visualize and optimize product development in various industry through product, process and organization DSMs. Many researchers have explored the application of DSM in software development [14], [22], [23]

2.2 Optimization of Allocated Resources

However, DSM sequence optimization is a NP-hard problem [24] and cannot be solved using simple optimization techniques. Yassine et al., have demonstrated how optimization can be performed on a multi-domain solution using a single objective function rather than optimizing individual domains separately [25]. They claim that, in doing so, more efficient multi-domain DSM arrangements may be discovered that were not initially apparent. Golfarelli et al., have proposed a framework to optimize the software development process to maximize value creation [26].

2.3 Opportunity and the Current Research

Though there is literature regarding the practices to estimate, plan, and execute in an agile environment, the tension between value, size, and dependencies in the overall value generation has not been studied extensively. Using the field study conducted at Swisscom as our motivation, we attempt to perform an exploratory study on the impact of effort, capacity, and dependency on value creation in agile sprints and propose a model that takes these factors into consideration to maximize the value creation by decoupling cyclic dependencies and optimizing the resource allocation.

3. Modeling Agile Value Creation

3.1 Objective

To explore value maximization in agile sprints, we formulate a stylized model with multiple features, two sprints, and two resources (persons). We have added further constraints in the form of dependencies between the various stories that belong to the features. Since a feature is considered as the entity that delivers business value to stakeholders, it is important that all stories in a feature are delivered to realize the value of a functionality. Sprint 1 and 2 are representational for modeling purposes. In reality, they may be considered as any two sprints that are consecutive.

Multi-Domain Matrix (MDM) can be a useful tool to gain insights into product development by visualizing the inter-relations between various domains – people, process, and product [25]. MDM is a form of DSM that represents multiple domains connected by various relationship types. Figure 3 shows a stylized MDM with two domains – people and process. In our model, we represent these two domains. The DSM contains the story level information and would give insights into the sequence in which stories must be completed. At the same time, the Domain mapping matrix (DMM) [27] includes the information regarding the resource allocation required to complete these stories. Hence, the DSM represents the process domain(process DSM), and DMM represents the people domain (resource DMM).

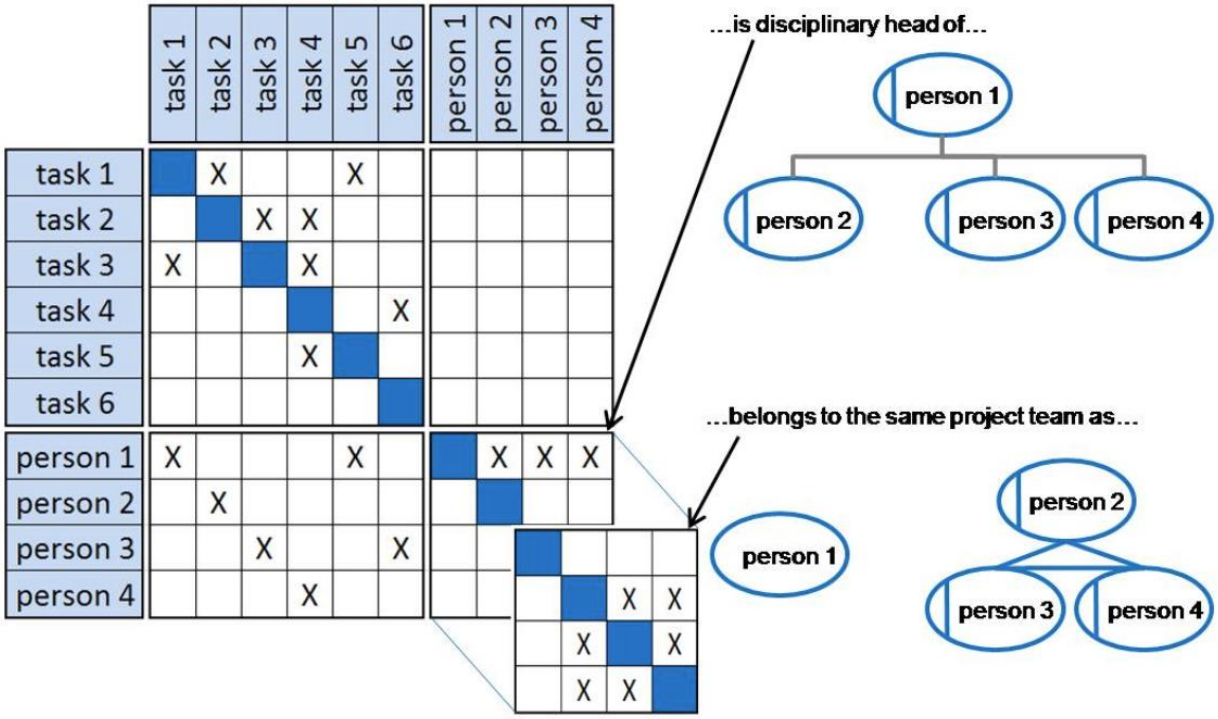

task 1
task 2
task 3
task 4
task 5
task 6
person 1
person 2
person 3
person 4
...is disciplinary head of...
...belongs to the same project team as...
X

Figure 3: Stylized MDM (source: https://dsmweb.org/multiple-domain-matrix-mdm/)

We summarize the objective of this research as follows:

1. Create most value achievable in the first sprint
2. Set up for additional value creation in the second sprint with limited lookahead

3.2 Sequencing Considerations

Extensive studies have demonstrated how value, effort, and resources impact the prioritization process. This is not limited to software development. WSJF is a popular model for prioritization for organizations that practice Agile methodology. However, it does not consider the role of dependencies. The role of dependency is a relatively less explored topic in agile software development [28]. Dependency necessitates coordination. If dependencies are not handled well, it creates stress in the team and leads to undesired delays. Identifying and formally registering these dependencies are an essential step in managing dependencies. The same inferences were drawn during our discussion with the Manager. Our model considers value, effort (story size), resources (personnel), and dependencies as the considerations while sequencing work and optimizing value creation.

3.2.1 Decoupling Approach

Our discussions with the Manager revealed that cyclic dependencies cause stress to the developers, Scrum Master, Release Train Engineers and Product Owners alike as it increases the coordination overhead. Hence, we decided to focus our attention on decoupling these cyclic dependencies. Once the cyclic dependencies are removed, we can represent the stories as a lower triangular DSM, which provides more flexibility to manage the execution of stories and the resource allocation.

Our initial proposal assumed that if dependencies are to be decoupled, a buffer amount of effort must be added to the stories engaged in that cyclic dependency. We assumed that 50% may be added to a story as the buffer amount of effort. However, the Manager pointed out that since the dependency management occurs during initial planning phases, they would prefer redistributing efforts among stories rather than adding buffer. In doing so, the overall size of the backlog (in terms of story points) can be kept the same. The below diagram is a simplified representation of this process:

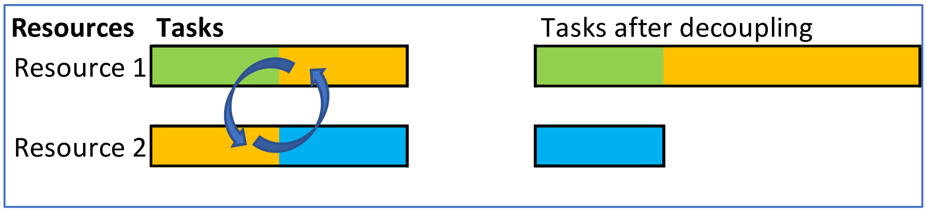


Figure 4: Simplified representation of decoupling process

In Figure 4, the portion of the tasks highlighted in Amber represents the portion that causes the cyclic dependency. By reassigning that portion to Resource 1, the cyclic dependency may be decoupled, reducing the necessity for coordination among the resources during execution.

Based on the feedback received and our analysis, we propose a strategy consisting of six methods to redistribute the efforts among stories to break cyclic dependencies. The effort redistributed accounts for the portion of work that causes this cyclic dependency. To simplify the modelling process, we have assumed that 50% of a story accounts for the effort contributing to that portion of work; however, once the framework has been programmatically implemented, this numerical value can be parameterized. The goal of this redistribution of efforts is to resequence this DSM as a lower triangular DSM.

Table 1 summarizes the Sequencing Strategies based on the above two parameters.

Table 1: Decoupling Methods

	Direction of Transfer	
Selection Criteria	**Small to Large**	**Large to Small**
Effort based while ignoring value	*Method 1*	*Method 2*
Address smaller value story/feature first	*Method 3*	*Method 4*
Address larger value story/feature first	*Method 5*	*Method 6*

The Selection Criteria column contains information regarding the criteria based on a method that would be selected and applied. Small to Large and Large to Small columns represent the direction of transfer of the efforts. Effort based methods (method 1 and method 2) consider only effort as a parameter for transfer. This may be applied when the dependencies occur within a feature or when the stories with cyclic dependencies belong to 2 features with equal value. The value-first methods (methods 3,4,5, and 6) consider value of the features as the first decision point followed by effort. This may occur when cyclic dependencies occur among stories belong to features with different values. For example, method 5 is a value-first method. If two stories have a cyclic dependency and belong to two features that have different values, identify the feature with higher value first, then transfer efforts from smaller story that belong to the higher value feature to larger story in that relationship. Additionally, if the transfer causes a story to exceed a single resource capacity limit for it to be completed within a sprint, the story must be split or broken down. The resultant stories will inherit the dependencies of the parent story.

We have applied these six methods independently to an arbitrary test dataset to understand how they impact the value creation in a PI. We do acknowledge that not all dependencies may be eliminated. However, decoupling cyclic dependencies to make all the dependencies sequential improves flexibility and enables better planning. These strategies are mere guidelines. It is the prerogative of the Scrum Master and Product Owner to decide how the efforts must be reassigned or transferred.

Test Dataset

Our test dataset represents a PI that consists of two sprints, with dependencies expressed as a DSM. The program has two persons to complete the four features (and eight stories). As there are only five days in a sprint, the total available capacity is 20 person-days. However, to complete the features listed, it would take 23 person-days, which implies that not all features will be completed. The steps below demonstrate how maximum value can be achieved given the constraints listed above. The ‘x’ in the cells of the DSM represents the dependency between the stories.

- PI – 2 sprints (5 days/sprint)
- Resources available - 2 persons
- Total Features – 4 (color-coded)

- Value defined at feature level; Max value available in the PI is 10
- Total Stories – 8 (2 per feature)
- Total effort required to complete all stories – 23 person days

Feature ID	Value	Story ID	Story point expressed in person days	S-11	S-10	S-20	S-21	S-30	S-31	S-40	S-41
F-1	1	S-10	1	■							
F-1		S-11	4		■						
F-2	2	S-20	4	x		■					
F-2		S-21	2				■	x			
F-3	3	S-30	2	x				■	x	x	
F-3		S-31	4					x	■	x	
F-4	4	S-40	2					x	x	■	
F-4		S-41	4	x	x			x	x		■

Figure 5: Test Dataset represented as a DSM

From the DSM, we can see that stories S-30, S-31, and S-40 have a cyclic dependency.

The goal of our model is to maximize the value that is generated in the first sprint while, with limited look ahead, maximize the value in the second sprint. We are considering the dependencies, value, and efforts as constraints. In the next section, we illustrate how value creation can be maximized by applying a strategy. For demonstration purposes, we have used transfer strategy 5 in the below illustration.

3.2.2 Illustration of the Modelling process

This section illustrates the process by which the bi-directional dependencies are decoupled, and the optimization is performed.

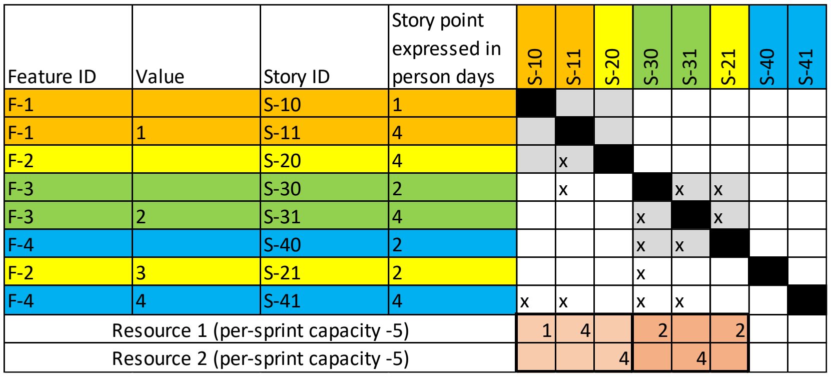
Step 1: Sequence the DSM; sequence the coupled blocks according to the efforts.

Feature ID	Value	Story ID	Story point expressed in person days	S-11	S-10	S-31	S-30	S-40	S-20	S-21	S-41
F-1		S-11	4	■							
F-1	1	S-10	1		■						
F-3		S-31	4			■	x	x			
F-3	2	S-30	2	x		x	■	x			
F-4		S-40	2			x	x	■			
F-2		S-20	4	x					■		
F-2	3	S-21	2				x			■	
F-4	4	S-41	4	x	x	x	x				■
Resource 1 (per-sprint capacity -5)				4			2	2			
Resource 2 (per-sprint capacity -5)					1	4			4		

▨ Sprint in process DSM
▨ Sprint 1 in people DMM
▨ Sprint 2 in people DMM

Figure 6: Dataset after initial sequencing

Though we achieve value in the first sprint, dependencies span across sprints. Hence, stories S-31, S-30, and S-40 cannot be completed. The cells highlighted in red in the resource DMM indicate the user stories that cannot be completed due to the dependencies.

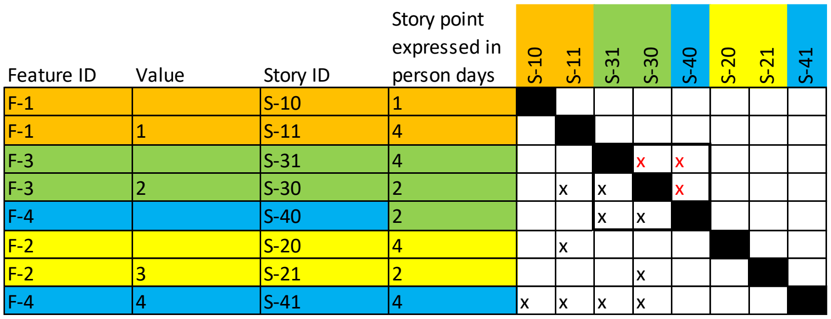


Feature ID	Value	Story ID	Story point expressed in person days	S-10	S-11	S-20	S-30	S-31	S-21	S-40	S-41
F-1		S-10	1	■							
F-1	1	S-11	4		■						
F-2		S-20	4		x	■					
F-3		S-30	2		x		■	x	x		
F-3	2	S-31	4				x	■	x		
F-4		S-40	2				x	x	■		
F-2	3	S-21	2				x			■	
F-4	4	S-41	4	x	x		x	x			■
Resource 1 (per-sprint capacity -5)				1	4		2		2		
Resource 2 (per-sprint capacity -5)						4		4			

Figure 7: Suboptimal solution without decoupling

Figure 7 is a suboptimal solution obtained by resequencing. Though the dependencies do not span across sprints, the most valuable feature isn't still delivered.

Step 2: Identify feedback dependencies that are cyclic



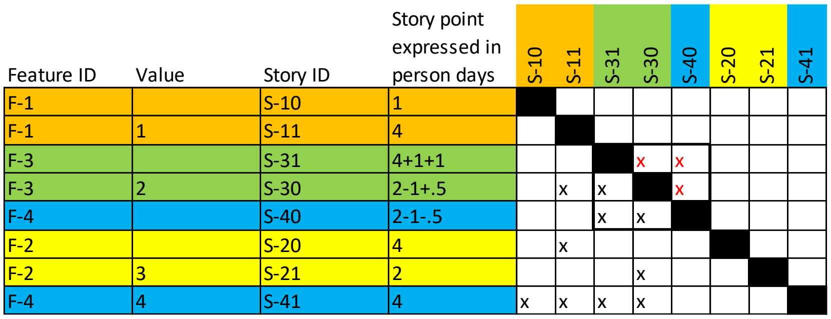
Feature ID	Value	Story ID	Story point expressed in person days	S-10	S-11	S-31	S-30	S-40	S-20	S-21	S-41
F-1		S-10	1	■							
F-1	1	S-11	4		■						
F-3		S-31	4			■	x	x			
F-3	2	S-30	2		x	x	■	x			
F-4		S-40	2			x	x	■			
F-2		S-20	4		x				■		
F-2	3	S-21	2				x			■	
F-4	4	S-41	4	x	x	x	x				■

Figure 8: DSM with coupled dependencies identified

x – feedback dependencies that are cyclic

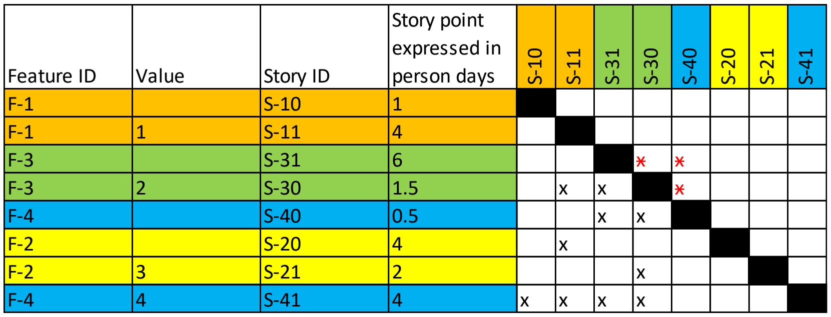
Step 3: Initiate transfer

Strategy 5 has been employed here for the purpose of demonstration. S-31 receives feedback from S-30 and S-40. Since S-31 is the largest story and F-4 has a higher value, the efforts are transferred to S-31.



Feature ID	Value	Story ID	Story point expressed in person days	S-10	S-11	S-31	S-30	S-40	S-20	S-21	S-41
F-1		S-10	1	■							
F-1	1	S-11	4		■						
F-3		S-31	4+1+1			■	x	x			
F-3	2	S-30	2-1+.5		x	x	■	x			
F-4		S-40	2-1-.5			x	x	■			
F-2		S-20	4		x				■		
F-2	3	S-21	2				x			■	
F-4	4	S-41	4	x	x	x	x				■

Figure 9: Step 1 in decoupling process. Efforts are transferred from feature with higher value to that with lower value from smaller story to larger story



Feature ID	Value	Story ID	Story point expressed in person days	S-10	S-11	S-31	S-30	S-40	S-20	S-21	S-41
F-1		S-10	1	■							
F-1	1	S-11	4		■						
F-3		S-31	6			■	~~x~~	~~x~~			
F-3	2	S-30	1.5		x	x	■	~~x~~			
F-4		S-40	0.5			x	x	■			
F-2		S-20	4		x				■		
F-2	3	S-21	2				x			■	
F-4	4	S-41	4	x	x	x	x				■

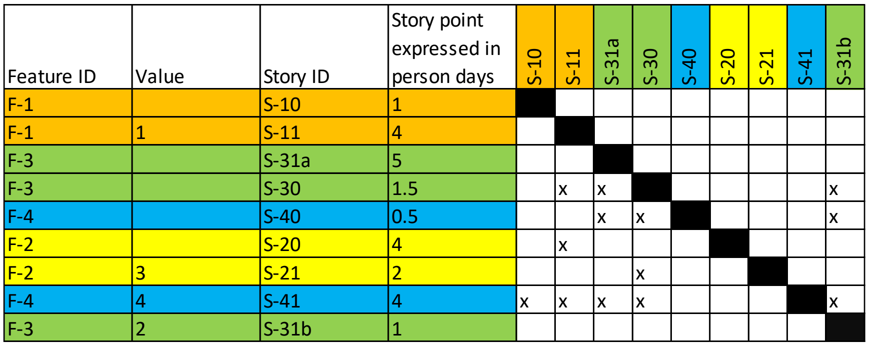
Figure 10: Stories with newly assigned efforts post decoupling

Step 4: Repeat Step 3 until all cyclic dependencies are decoupled

In Figure 10, the process has been repeated until all three feedback dependencies are decoupled in the process DSM.

Step 5: If a newly estimated story exceeds the resource capacity limit, split it into two

It can be seen that S-31 exceeds the person-days allocated to a sprint for a story. Hence it has to be split into two, S-31a and S-31b. Splitting a story may improve its clarity and make it more concrete [29]. The resultant DSM is:

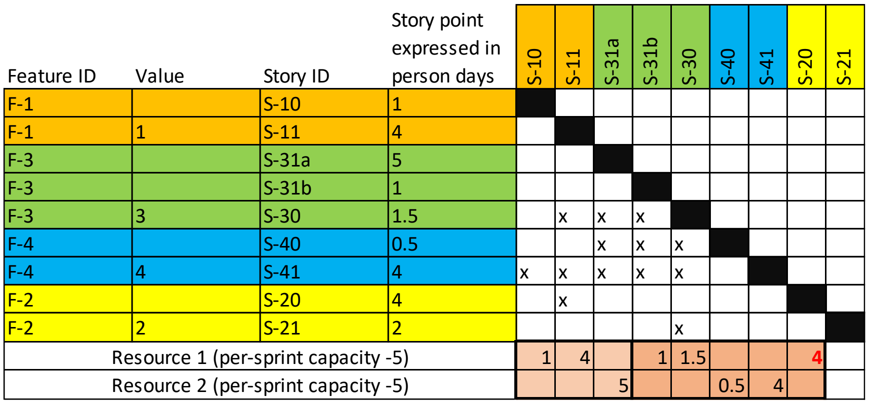


Feature ID	Value	Story ID	Story point expressed in person days	S-10	S-11	S-31a	S-30	S-40	S-20	S-21	S-41	S-31b
F-1		S-10	1	■								
F-1	1	S-11	4		■							
F-3		S-31a	5			■						
F-3		S-30	1.5		x	x	■					x
F-4		S-40	0.5			x	x	■				x
F-2		S-20	4		x				■			
F-2	3	S-21	2				x			■		
F-4	4	S-41	4	x	x	x	x				■	x
F-3	2	S-31b	1									■

Figure 11: DSM with the additional story after splitting

The newly created stories retain the dependencies of their parent story. In this case, S-31a and S-31b will have the same dependencies as S-31.

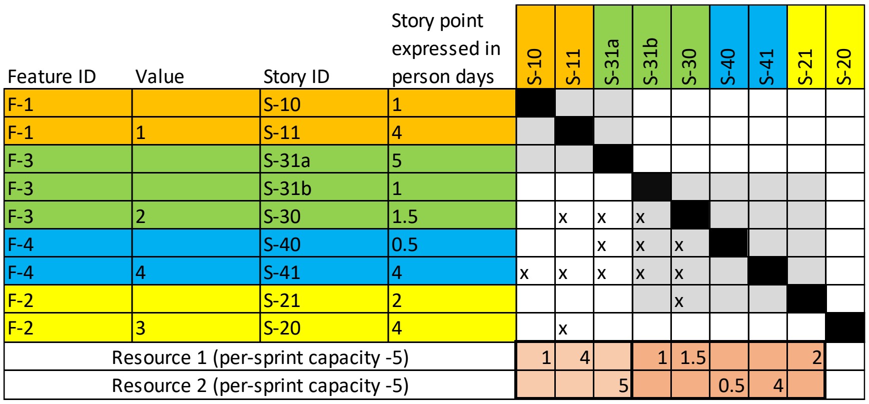
Step 6. Repeat Step 1



Feature ID	Value	Story ID	Story point expressed in person days	S-10	S-11	S-31a	S-31b	S-30	S-40	S-41	S-20	S-21
F-1		S-10	1	■								
F-1	1	S-11	4		■							
F-3		S-31a	5			■						
F-3		S-31b	1				■					
F-3	3	S-30	1.5		x	x	x	■				
F-4		S-40	0.5			x	x	x	■			
F-4	4	S-41	4	x	x	x	x	x		■		
F-2		S-20	4		x						■	
F-2	2	S-21	2					x				■
Resource 1 (per-sprint capacity -5)				1	4		1	1.5			4	
Resource 2 (per-sprint capacity -5)						5			0.5	4		

Figure 12: Re-sequenced DSM

Step 7. Run value maximization optimizer



Feature ID	Value	Story ID	Story point expressed in person days	S-10	S-11	S-31a	S-31b	S-30	S-40	S-41	S-21	S-20
F-1		S-10	1	■								
F-1	1	S-11	4		■							
F-3		S-31a	5			■						
F-3		S-31b	1				■					
F-3	2	S-30	1.5		x	x	x	■				
F-4		S-40	0.5			x	x	x	■			
F-4	4	S-41	4	x	x	x	x	x		■		
F-2		S-21	2					x			■	
F-2	3	S-20	4		x							■
Resource 1 (per-sprint capacity -5)				1	4		1	1.5			2	
Resource 2 (per-sprint capacity -5)						5			0.5	4		

Figure 13: MDM after optimization with resource allocation

The above PI plan promises maximum value without violating the constraints. It must be noted that the resource allocation is random and does not take a resource's attributes such as skillset and availability into account.

- Sprints completed - 2
- Total resources utilized - 19 person-days
- Total features completed – 3 out of 4
- The maximized value created in sprint 1 subject to constrain - 1
- **Total Value achieved – 8 out of 10**
- Total stories completed – 8 out of 9
- Sprint 1 utilization – 100%
- Sprint 2 utilization – 90%

3.2.3 Mathematical representation of the Optimization approach

We have defined an optimization strategy for the assignment of resources to maximize value creation. This section describes the input parameters and notations, objective function, and the constraints to perform the optimization. We use two domains – process and people – in our optimization process.

Input & Notation

Input	Notation
# of sprints – set to 2	Sprint 1 when t<=5, sprint 2 when 6<=t<=10 ; t is an integer; $t \in [1,10]$
Daily resource limit on capacity (# of persons) – assuming that the team size is fixed	$D \; \forall \; t$
# features that need to get worked on	F
Each feature's value when completed	$M_f \quad f \in [1,F]$; f is an integer
S_i is the identification number of a story	S_i is an integer; $i \in [1, S_{max}]$
Total # of stories	S_{max}
Each story is uniquely associated with one feature	Set: ={subset 1, subset 2…subset f}
Story point expressed in (p) person days	For each S_i, there is a p_i

Binary decision variable:

- r_{it} = resource allocated to a Story, S_i at time t

Objective function:

- $\max\limits_{r_i} (\Sigma V_{f,t \text{ at } t=5,10})$ …(1)

Constraints:

- Daily resource constraint
  The daily resource constraint implies that resources allocated in a given day must not exceed the total capacity available for that day. For example, if there are 2 resources, D=2 and $\Sigma r_{it} \leq 2$

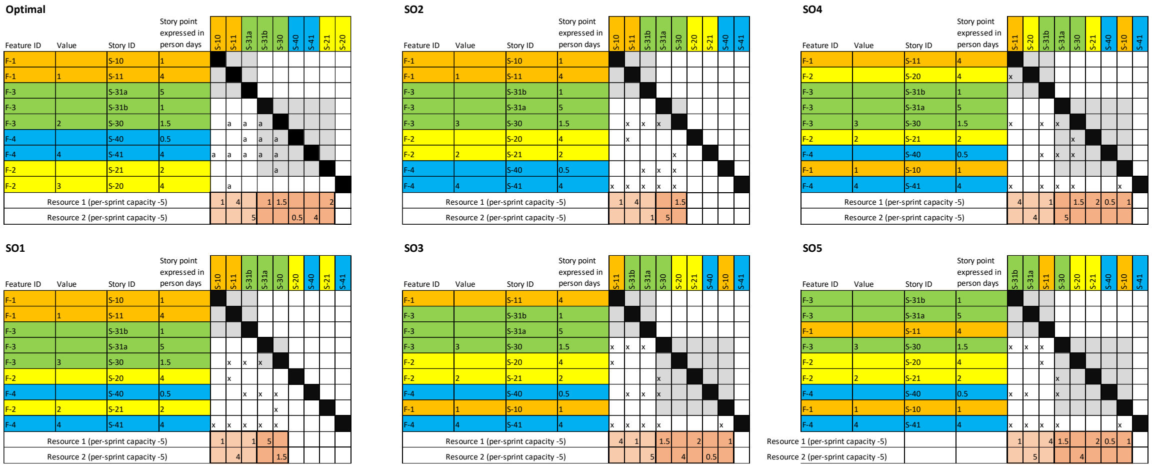
  $$D \geq \sum_{t=1}^{T} r_{it} \text{ , } \forall t \qquad \ldots (2)$$
- Value creation constraint
  Value creation constraint implies that V will be generated only when all stories of a feature are complete.

  $$V_{f,t} = M_f \text{, when for subset f, } \sum_{t=1}^{T} p_i = \sum_{t=1}^{T} r_{it} \text{ , } \forall t$$

  $$\text{else} = 0 \text{, when for subset f, } \sum_{t=1}^{T} pi > \sum_{t=1}^{T} r_{it} \text{, } \forall t \qquad \ldots(3)$$

3.2.4 Optimization by Manual Enumeration

In this thesis, we have performed the optimization through manual enumeration. For the above set of nine user stories 9! Or 3,62,880 sequences are possible. However, if we consider the dependencies present in the test dataset, we can reduce the solution space to 33 sequences. These 33 sequences abide by the constraint that the resultant DSM that has been subject to Sequencing Strategy must be lower triangular DSM. Here are a few suboptimal solutions that we obtained during the manual enumeration process without violating the dependency constraints. In Figure 14, SO1 to SO5 are five suboptimal solutions that were randomly selected.



Optimal

Feature ID	Value	Story ID	Story point expressed in person days	S-10	S-11	S-31a	S-31b	S-30	S-40	S-41	S-21	S-20
F-1		S-10	1	■								
F-1	1	S-11	4		■							
F-3		S-31a	5			■						
F-3		S-31b	1				■					
F-3	2	S-30	1.5		a	a	a	■				
F-4		S-40	0.5			a	a	a	■			
F-4	4	S-41	4	a	a	a	a	a		■		
F-2		S-21	2					a			■	
F-2	3	S-20	4		a							■
Resource 1 (per-sprint capacity -5)				1	4		1	1.5			2	
Resource 2 (per-sprint capacity -5)						5			0.5	4		

SO2

Feature ID	Value	Story ID	Story point expressed in person days	S-10	S-11	S-31b	S-31a	S-30	S-20	S-21	S-40	S-41
F-1		S-10	1	■								
F-1	1	S-11	4		■							
F-3		S-31b	1			■						
F-3		S-31a	5				■					
F-3	3	S-30	1.5		x	x	x	■				
F-2		S-20	4		x				■			
F-2	2	S-21	2					x		■		
F-4		S-40	0.5			x	x	x			■	
F-4	4	S-41	4	x	x	x	x	x				■
Resource 1 (per-sprint capacity -5)				1	4			1.5				
Resource 2 (per-sprint capacity -5)						1	5					

SO4

Feature ID	Value	Story ID	Story point expressed in person days	S-11	S-20	S-31b	S-31a	S-30	S-21	S-40	S-10	S-41
F-1		S-11	4	■								
F-2		S-20	4	x	■							
F-3		S-31b	1			■						
F-3		S-31a	5				■					
F-3	3	S-30	1.5	x		x	x	■				
F-2	2	S-21	2					x	■			
F-4		S-40	0.5			x	x	x		■		
F-1	1	S-10	1								■	
F-4	4	S-41	4	x		x	x	x			x	■
Resource 1 (per-sprint capacity -5)				4		1		1.5	2	0.5	1	
Resource 2 (per-sprint capacity -5)					4		5					

SO1

Feature ID	Value	Story ID	Story point expressed in person days	S-10	S-11	S-31b	S-31a	S-30	S-20	S-40	S-21	S-41
F-1		S-10	1	■								
F-1	1	S-11	4		■							
F-3		S-31b	1			■						
F-3		S-31a	5				■					
F-3	3	S-30	1.5		x	x	x	■				
F-2		S-20	4		x				■			
F-4		S-40	0.5			x	x	x		■		
F-2	2	S-21	2					x			■	
F-4	4	S-41	4	x	x	x	x	x				■
Resource 1 (per-sprint capacity -5)				1		1	5					
Resource 2 (per-sprint capacity -5)					4			1.5				

SO3

Feature ID	Value	Story ID	Story point expressed in person days	S-11	S-31b	S-31a	S-30	S-20	S-21	S-40	S-10	S-41
F-1		S-11	4	■								
F-3		S-31b	1		■							
F-3		S-31a	5			■						
F-3	3	S-30	1.5	x	x	x	■					
F-2		S-20	4	x				■				
F-2	2	S-21	2				x		■			
F-4		S-40	0.5		x	x	x			■		
F-1	1	S-10	1								■	
F-4	4	S-41	4	x	x	x	x				x	■
Resource 1 (per-sprint capacity -5)				4	1		1.5		2		1	
Resource 2 (per-sprint capacity -5)						5		4		0.5		

SO5

Feature ID	Value	Story ID	Story point expressed in person days	S-31b	S-31a	S-11	S-30	S-20	S-21	S-40	S-10	S-41
F-3		S-31b	1	■								
F-3		S-31a	5		■							
F-1		S-11	4			■						
F-3	3	S-30	1.5	x	x	x	■					
F-2		S-20	4			x		■				
F-2	2	S-21	2				x		■			
F-4		S-40	0.5	x	x		x			■		
F-1	1	S-10	1								■	
F-4	4	S-41	4	x	x	x	x				x	■
Resource 1 (per-sprint capacity -5)				1		4	1.5		2	0.5	1	
Resource 2 (per-sprint capacity -5)					5			4				

Figure 14: Examples of optimal and suboptimal results obtained through manual enumeration

Table 2: Comparison of suboptimal and optimal solutions

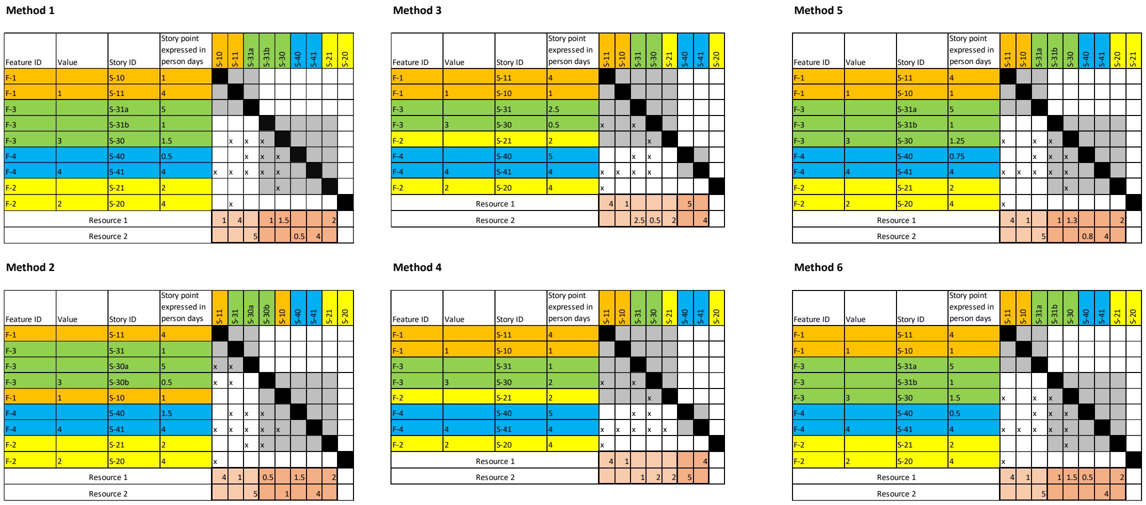
	$V_{sprint1}$	$V_{sprint2}$	V_{total}	$Util_{sprint1}$	$Util_{sprint2}$
Optimal	1	7	8	100%	90%
SO1	1	3	4	60%	65%
SO2	1	3	4	60%	65%
SO3	0	6	6	100%	90%
SO4	0	6	6	90%	100%
SO5	0	6	6	100%	90%

Table 2 summarizes the efficacy of these solutions. It is evident from the table that the suboptimal solution could lead to the underutilization of resources and reduce value creation.

4. Discussion and Analysis

4.1 Decouple Method Comparison

Figure 15 shows the MDM representation of the PI that has been independently optimized using the six decoupling methods. It is evident from the below images that the stories that are planned for a sprint change significantly with the method chosen. The MDM provides an easy way to visualize the PI plan and the resource allocation simultaneously.



Method 1

Feature ID	Value	Story ID	Story point expressed in person days	S-10	S-11	S-31a	S-31b	S-30	S-40	S-41	S-21	S-20
F-1		S-10	1	■								
F-1	1	S-11	4		■							
F-3		S-31a	5			■						
F-3		S-31b	1				■					
F-3	3	S-30	1.5		x	x	x	■				
F-4		S-40	0.5			x	x	x	■			
F-4	4	S-41	4	x	x	x	x	x		■		
F-2		S-21	2					x			■	
F-2	2	S-20	4		x							■
Resource 1				1	4		1	1.5			2	
Resource 2						5			0.5	4		

Method 2

Feature ID	Value	Story ID	Story point expressed in person days	S-11	S-31	S-30a	S-30b	S-10	S-40	S-41	S-21	S-20
F-1		S-11	4	■								
F-3		S-31	1		■							
F-3		S-30a	5	x	x	■						
F-3	3	S-30b	0.5	x	x		■					
F-1	1	S-10	1					■				
F-4		S-40	1.5		x	x	x		■			
F-4	4	S-41	4	x	x	x	x	x		■		
F-2		S-21	2			x	x				■	
F-2	2	S-20	4	x								■
Resource 1				4	1		0.5		1.5		2	
Resource 2						5		1		4		

Method 3

Feature ID	Value	Story ID	Story point expressed in person days	S-11	S-10	S-31	S-30	S-21	S-40	S-41	S-20
F-1		S-11	4	■							
F-1	1	S-10	1		■						
F-3		S-31	2.5			■					
F-3	3	S-30	0.5	x		x	■				
F-2		S-21	2				x	■			
F-4		S-40	5			x	x		■		
F-4	4	S-41	4	x	x	x	x			■	
F-2	2	S-20	4	x							■
Resource 1				4	1				5		
Resource 2						2.5	0.5	2		4	

Method 4

Feature ID	Value	Story ID	Story point expressed in person days	S-11	S-10	S-31	S-30	S-21	S-40	S-41	S-20
F-1		S-11	4	■							
F-1	1	S-10	1		■						
F-3		S-31	1			■					
F-3	3	S-30	2	x		x	■				
F-2		S-21	2				x	■			
F-4		S-40	5			x	x		■		
F-4	4	S-41	4	x	x	x	x	x		■	
F-2	2	S-20	4	x							■
Resource 1				4	1					4	
Resource 2						1	2	2	5		

Method 5

Feature ID	Value	Story ID	Story point expressed in person days	S-11	S-10	S-31a	S-31b	S-30	S-40	S-41	S-20	S-21
F-1		S-11	4	■								
F-1	1	S-10	1		■							
F-3		S-31a	5			■						
F-3		S-31b	1				■					
F-3	3	S-30	1.25	x		x	x	■				
F-4		S-40	0.75			x	x	x	■			
F-4	4	S-41	4	x	x	x	x	x		■		
F-2		S-21	2					x			■	
F-2	2	S-20	4	x								■
Resource 1				4	1		1	1.3			2	
Resource 2						5			0.8	4		

Method 6

Feature ID	Value	Story ID	Story point expressed in person days	S-11	S-10	S-31a	S-31b	S-30	S-40	S-41	S-21	S-20
F-1		S-11	4	■								
F-1	1	S-10	1		■							
F-3		S-31a	5			■						
F-3		S-31b	1				■					
F-3	3	S-30	1.5	x		x	x	■				
F-4		S-40	0.5			x	x	x	■			
F-4	4	S-41	4	x	x	x	x	x		■		
F-2		S-21	2					x			■	
F-2	2	S-20	4	x								■
Resource 1				4	1		1	1.5	0.5		2	
Resource 2						5				4		

Figure 15: Optimized MDMs using six decoupling methods

We have evaluated these methods to see how value created differs when applied to the same dataset by analyzing the various measures of central tendency.

Here's a summary of the results achieved using each method.

Table 3: Comparison of results obtained using the six methods

Strategy ID	Strategy desc	Mean story size	Std dev of story size	Median story size	$V_{sprint1}$	$V_{sprint2}$	Total Value	$Util_{sprint1}$	$Util_{sprint2}$
1	Transfer from smaller story to larger story	2.6	1.69	2	1	7	8	100%	90%
2	Transfer from larger story to smaller story	2.6	1.69	2	0	8	8	100%	90%
3	Transfer from smaller value feature to larger value feature + Transfer from smaller story to larger story	2.9	1.56	3.25	4	4	8	100%	90%
4	Transfer from lower value feature to higher value feature + Transfer from larger story to smaller story	2.9	1.62	3	4	4	8	100%	90%
5	Transfer from higher value feature to lower value feature + Transfer from smaller story to larger story	2.6	1.67	2	1	7	8	100%	90%
6	Transfer from higher value feature to lower value feature + Transfer from larger story to smaller story	2.6	1.69	2	1	7	8	100%	90%

From the above summary table, we can infer that the choice of decoupling strategy impacts the overall speed of delivery of value while maintaining the same resource utilization. By simultaneously managing dependencies and resource allocation, we can alter the value creation in a sprint.

Since the methods have been applied only to a small dataset, we cannot draw a more generalized conclusion. However, if applied to a larger dataset, we hope to draw more realistic conclusions regarding the efficacy of the decoupling strategy. It must be noted that these methods may not always be applicable interchangeably and must be picked based on the situation. However, using a consistent strategy may help in creating a standard practice in the team.

4.2 Managerial insights from Swisscom

Manager indicates Strategy 3 is better than Strategy 4 as it has a lower Standard Deviation for story size. This is considered to be the best practice in the agile world because it keeps the work balanced. Based on the feedback we received, we came to the following abductive hypothesis.

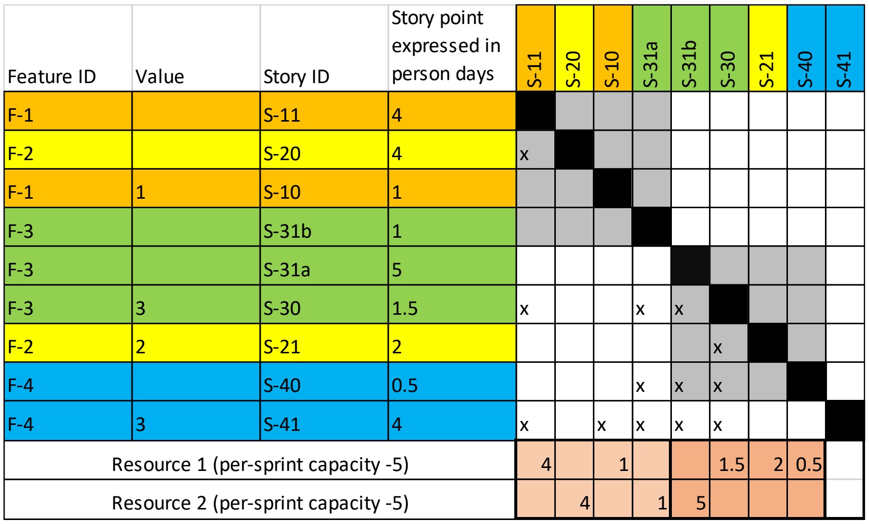
Hypothesis: If the size of the story is capped at a pre-defined sprint limit while lowering the standard deviation, more flexibility can be achieved, enabling higher value creation

Manager also expressed his interest in knowing the effect of time criticality and assignment of value at the story level on the framework. We have demonstrated the impact in the next section – special cases.

4.3 Special cases

4.3.1 Time Criticality

Though we haven’t considered time criticality as a parameter in the model, we are attempting to demonstrate how the output of the model will vary when it is factored it into our test. A feature that is considered by a team as less valuable may have a dependency with a feature of higher value on another team’s backlog within the organization. This may occur due to siloed planning, lack of clear understanding of features at an organization level, among other reasons. In the example below, we have considered Feature F-2 as a time-critical feature that must be completed before the end of PI, although it is not the most valuable feature. It can be seen that the most valuable feature has not been completed due to capacity constraints. Hence, we can infer that there exists a tradeoff between time criticality and value generated.



Feature ID	Value	Story ID	Story point expressed in person days	S-11	S-20	S-10	S-31a	S-31b	S-30	S-21	S-40	S-41
F-1		S-11	4	■	▨	▨	▨					
F-2		S-20	4	x	■	▨	▨					
F-1	1	S-10	1	▨	▨	■	▨					
F-3		S-31b	1	▨	▨	▨	■					
F-3		S-31a	5					■	▨	▨	▨	
F-3	3	S-30	1.5	x			x	x	■	▨	▨	
F-2	2	S-21	2					▨	x	■	▨	
F-4		S-40	0.5				x	x	x	▨	■	
F-4	3	S-41	4	x		x	x	x	x			■
Resource 1 (per-sprint capacity -5)				4		1			1.5	2	0.5	
Resource 2 (per-sprint capacity -5)					4		1	5				

Figure 16: Optimized solution for Special case with Time Critical feature

4.3.2 Assigning Value at Story Level

This is a special case where value is assigned to the stories instead of features. According to the Manager, the best practice for a Product Owner is to identify and assign value to each story as opposed to assigning value to features. This can be considered a special case in our model where each feature has only one story. Hence, the value and story will have a one-to-one relationship. However, to provide readers the ability to compare all the models, instead of creating a new example dataset, we have distributed the value of the features to its stories so that the overall value remains the same, and so does the effort to complete each story.

Here is the DSM from Step 7 in section <>. The process remains the same until this point. The below DSM has the value represented at the story level. The value of the feature has been distributed. However, it can be noticed that though the feature value may be lower, its stories may have a relatively higher value when compared to stories belonging to higher-value features. For example, feature F-4 had a value of 4, whereas F-2 has only 2, which implies that it is desirable to complete F-4 before F-2. But after defining value at story-level, Story S-21, with a value of 1.5, which belongs to F-2 has become more valuable than S-40 that belongs to F-4 as S-40 has only a value of 1.

Feature ID	Value	Story ID	Story point expressed in person days	S-11	S-20	S-10	S-31a	S-31b	S-30	S-21	S-40	S-41
F-1	0.5	S-11	4	■								
F-2	0.5	S-20	4	x	■							
F-1	0.5	S-10	1			■						
F-3	0.5	S-31b	1				■					
F-3	2	S-31a	5					■				
F-3	0.5	S-30	1.5	x			x	x	■			
F-2	1.5	S-21	2						x	■		
F-4	1	S-40	0.5				x	x	x		■	
F-4	3	S-41	4	x		x	x	x	x			■

Figure 17: Special case where value is assigned at story level

After optimizing the MDM, the result indicates that the total value achieved is more than what was achieved when value was defined at the feature level despite spending the same amount of effort. Hence, it may be inferred that determining value at the story level may improve the perceived value creation in a PI.

Feature ID	Value	Story ID	Story point expressed in person days	S-31b	S-31a	S-11	S-10	S-30	S-21	S-41	S-40	S-20
F-3	0.5	S-31b	1	■								
F-3	0.5	S-31a	5		■							
F-1	2	S-11	4			■						
F-1	0.5	S-10	1				■					
F-3	0.5	S-30	1.5	x	x	x		■				
F-2	1.5	S-21	2					x	■			
F-4	3	S-41	4	x	x	x	x	x		■		
F-4	1	S-40	0.5	x	x			x			■	
F-2	0.5	S-20	4			x						■
Resource 1 (per-sprint capacity -5)				1		4	1	1.5	2			
Resource 2 (per-sprint capacity -5)					5					4	0.5	

Figure 18: Optimized solution for the Special case where value is assigned at story level

5. Conclusion and Direction for Future Work

In this thesis, we conducted initial fieldwork at Swisscom to understand some of the critical challenges in scaled agile software development. Based on the discussions with the Manager, we identified the creation of value earlier in the PI as a critical challenge and research opportunity. A key factor contributing to this challenge is bi-directional dependencies between engineers stemming from the technical coupling between user stories. At Swisscom, unless a feature is completed, stakeholders do not get to realize the value of the deliverable or functionality. Though not all dependencies may be eliminated, it is essential that they are simplified and managed effectively. We identified DSM as an effective tool to help decouple the dependencies and MDM as a tool to help represent the resource allocation and sequence of tasks simultaneously.

We analyzed six decoupling methods using a test dataset and reviewed the results with the Manager. We relied on the results to demonstrate the efficacy of our framework. We also drew an abductive hypothesis, which was tested using the framework. We proposed an optimization approach to maximize value creation by simultaneously managing dependencies and resource allocation that we demonstrated through the manual enumeration process.

Limitations and Opportunities for Future Work:

However, there are many limitations to the approach presented in the thesis. The model itself makes several simplifying assumptions in decoupling the dependencies. Firstly, the amount of effort that is redistributed, 50%, may not be a realistic representation of the effort that contributes to the cyclic dependency. It was a figure selected purely for representational purposes. It may vary based on several factors. Additionally, the model assumes that the stories become entirely decoupled when the efforts are redistributed, which may not always be accurate. It may depend on factors such as clarity in requirements, technical complexity, and resource constraints, to name a few.

Eppinger and Browning [21] have suggested several methods to resolve the coupling that may serve as an alternative or supplement to the decoupling methods presented in this thesis. In this thesis, we have combined two of these methods to create the proposed strategy. The tearing method mentioned in the book, which was initially presented by Steward [30], explains how a block of coupled dependencies can be broken down into sequential dependencies with minimal iteration. The process involves representing these blocks as node-link circuits and tearing the link that breaks most or longest circuits. The tearing method suggests that these tears be accepted by the process owner based on knowledge of the process. This tearing of these links is based on certain assumptions, and in this thesis, we have assumed that by redistributing the portion that causes the dependency, we can decouple the stories.

Another decoupling method suggested by Eppinger & Browning that has been leveraged is Further Decomposition. Further decomposition may help identify less coupled stories as coupled stories are an aggregation of smaller activities or requirements. The extent to which these methods have been used in this thesis is limited, and we believe that further research on the application of these methods in managing the dependencies may yield better strategies for decoupling bi-directional dependencies. There are several other methods such as Simulation, Eigenstructure, Signal flow graphs and Markov chain that may be useful in an Agile context and requires further research.

The testing has been performed on a small dataset that may not represent all the constraints and challenges in a real set of user stories. According to our tests, Strategy 3 seems to deliver the best results, which may not always hold true. Hence these inferences must be validated using a larger dataset to measure the efficacy of the framework. The framework does not eliminate the need for planning altogether. It is a tool to assist the team in improving their planning. There are steps that the teams must take to identify the dependencies, value, and estimate the story as they are the inputs to the model. Like any model, unless the inputs are robust, the output may not represent reality. We identified two special cases based on the feedback received from Swisscom, which is not exhaustive. There may be additional cases that may or may not work with the current strategies. In which case, these strategies may need revision.

The test data consists of only three user stories with cyclic dependency. Since our model considers two user stories in a given step to decouple these dependencies, the decoupling process is theoretically scalable. A PI may consist of several hundred user stories, and more than two sprints. Hence the probability of having more complex bi-directional dependencies are higher. The model will need to be tested with real datasets to measure its efficacy, and more heuristics may be needed to resolve these dependencies.

Our optimization approach uses dependencies, value, and resource availability to define the objective function and constraints. However, during our research, we identified other methods that may be used for clustering and partitioning [31], [32], [33]. Further research and empirical studies are needed to identify a good optimization approach.

The skillset of the team and its individuals has not been taken into account in this thesis. However, it plays an essential role in the assignment of tasks. This could be factored in as an additional constraint to the objective function and needs further investigation.

6. References

[1] J. Manas, “Agile Project Management : What ’ s the story ?,” 2017, [Online]. Available: https://sistemas.uniandes.edu.co/images/forosisis/foros/fisw1/1er-Foro-ISW-presentacion-7.pdf.

[2] K. Beck *et al.*, “Manifesto for Agile Software Development,” *The Agile Alliance*, 2001. .

[3] D. Turk, R. France, and B. Rumpe, “Assumptions Underlying Agile Software Development Processes,” *Proceedings - Agile 2008 Conference*, vol. 16, no. 4, pp. 62–87, 2005.

[4] “Enterprise challenges How SAFe empowers businesses to compete in today’s marketplace.” https://www.scaledagile.com/enterprise-solutions/enterprise-challenges/.

[5] KPMG, “Agile Transformation 2019,” 2019, [Online]. Available: https://assets.kpmg/content/dam/kpmg/be/pdf/2019/11/agile-transformation.pdf.

[6] “ScaledAgileFramework.” https://www.scaledagileframework.com.

[7] K. Thompson, *Solutions for Agile Governance in the Enterprise (Sage)*. 2019.

[8] “Scrum@Scale.” https://scrumatscale.scruminc.com/.

[9] M. Cruth, “Discover the Spotify model.” https://www.atlassian.com/agile/agile-at-scale/spotify.

[10] “LeSS Framework.” https://less.works/less/framework/index.

[11] S. Ambler, “Disciplined Agile Delivery The Foundation for Scaling Agile.”

[12] L. Crofoot, “Management of Cross-Team Interfaces in Large-Scale Agile Development,” MIT, 2020.

[13] Swisscom, “Swisscom Company Profile.” https://www.swisscom.ch/en/about/company/portrait/profil.html.

[14] S. Bajpai, “Planning Large-Scale Agile Development Using a Dependency Structure Mapping Model,” MIT, 2020.

[15] D. Reinertsen, “The Principles of Product Development Flow: Second Generation Lean Product Development by Donald G. Reinertsen,” *Journal of Product Innovation Management*, vol. 27, no. 1, pp. 137–139, 2009, doi: 10.1111/j.1540-5885.2009.00705_1.x.

[16] M. Poppendieck, “Principles of lean thinking,” *IT Management Select*, pp. 1–7, 2011, [Online]. Available: http://world-scholarships.com/books/Books at LMDA/Lean Manufacturing/Poppendieck, Mary - Principles of Lean Thinking (2002, 7p).pdf.

[17] C. J. Torrecilla-Salinas, J. Sedeño, M. J. Escalona, and M. Mejías, “Estimating, planning

and managing Agile Web development projects under a value-based perspective," *Information and Software Technology*, vol. 61, pp. 124–144, 2015, doi: 10.1016/j.infsof.2015.01.006.

[18] B. Wake, "INVEST in Good Stories, and SMART Tasks," 2003. .

[19] A. Herrmann and M. Daneva, "Requirements Prioritization Based on Benefit and Cost Prediction: An Agenda for Future Research," pp. 125–134, 2008.

[20] E. Babinet and R. Ramanathan, "Dependency management in a large agile environment," 2008, doi: 10.1109/Agile.2008.58.

[21] S. D. Eppinger and T. R. Browning, *Design Structure Matrix Methods and Applications*. MIT Press, 2012.

[22] M. Stevens, "Design Structure Matrices for Software Development," *Faculteit Wetenschappen Vakgroep Computerwetenschappen Design*, 2007.

[23] N. Sangal, E. Jordan, V. Sinha, and D. Jackson, "Using dependency models to manage complex software architecture," *ACM SIGPLAN Notices*, vol. 40, no. 10, pp. 167–176, 2005, doi: 10.1145/1103845.1094824.

[24] M. R. Garey, D. S. Johnson, and L. Stockmeyer, "Some simplified NP-complete graph problems," *Theoretical Computer Science*, 1976, doi: 10.1016/0304-3975(76)90059-1.

[25] A. A. Yassine, R. H. Chidiac, and I. H. Osman, "Simultaneous optimisation of products, processes, and people in development projects," *Journal of Engineering Design*, vol. 24, no. 4, pp. 272–292, 2013, doi: 10.1080/09544828.2012.727206.

[26] M. Golfarelli, S. Rizzi, and E. Turricchia, "Multi-sprint planning and smooth replanning: An optimization model," *Journal of Systems and Software*, vol. 86, no. 9, pp. 2357–2370, 2013, doi: 10.1016/j.jss.2013.04.028.

[27] "Domain Mapping Matrix (DMM)." https://dsmweb.org/domain-mapping-matrix-dmm/.

[28] A. Martakis and M. Daneva, "Handling requirements dependencies in agile projects: A focus group with agile software development practitioners," *Proceedings - International Conference on Research Challenges in Information Science*, 2013, doi: 10.1109/RCIS.2013.6577679.

[29] O. Liskin, R. Pham, S. Kiesling, and K. Schneider, "Why we need a granularity concept for user stories," *Lecture Notes in Business Information Processing*, vol. 179 LNBIP, pp. 110–125, 2014, doi: 10.1007/978-3-319-06862-6_8.

[30] D. V. Steward, "Design Structure System: a Method for Managing the Design of Complex Systems.," *IEEE Transactions on Engineering Management*, vol. EM-28, no. 3, pp. 71–74, 1981, doi: 10.1109/TEM.1981.6448589.

[31] B. G. Cameron, "Value Network Modeling," MIT, 2007.

[32] E. Feng, Wen;F. Crawley, "Dependency structure matrix modelling for stakeholder value

networks," *Proceedings of the 12th International DSM Conference, Cambridge, UK*, vol. 501, no. November 2006, pp. 2007–2007, 2010.

[33] X. Yao, J. Zhou, Y. Li, and E. Liu, "An Enhanced Collaborative Optimization Approach with Design Structure Matrix Algorithms to Group and Decouple Multidisciplines," *Mathematical Problems in Engineering*, vol. 2016, 2016, doi: 10.1155/2016/4340916.