Analytics Driving Supply Chain Segmentation for Lenovo

Research Fest Presentation

Luiz Gosling & Javier Urrutia

MIT SCM Capstone Project

May 21th, 2019

Content

Lenovo operates 3 global, independent P&Ls…

Lenovo

Computing

Biggest revenue stream
$20B in 2017

Mobility

Motorola acquired from Google in 2016

Data Center (DCG)

Fairly new (3~4 y.o.)
$4B in 2017, $6.2B in 2018(E)

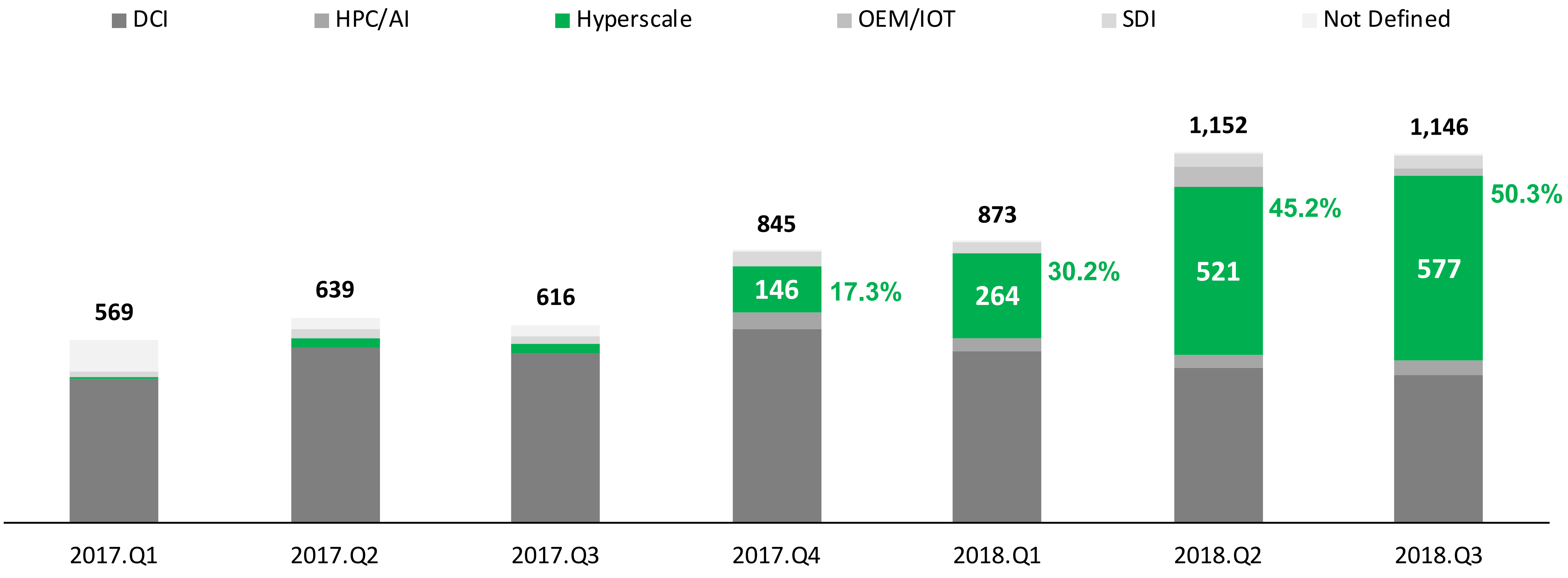
Cloud	DCI Storage	Performance
Telecom	IoT	Software

Source: Lenovo DCG, MIT CTL analysis.

What is a “data center solution”?

DCG's Hyperscale growth is driving Lenovo's rapid growth

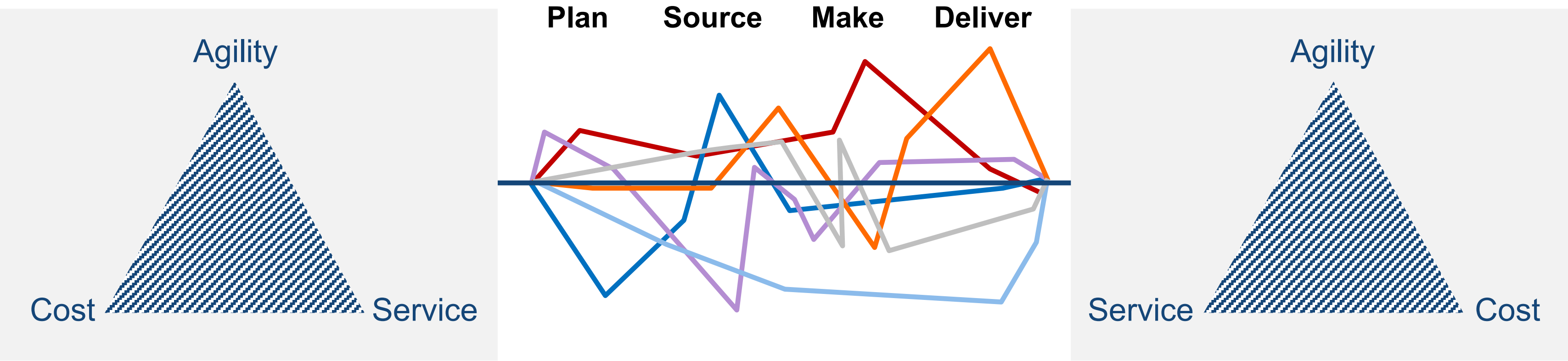
DCG Sales Breakdown and Evolution in North America
($ millions)



Source: Lenovo DCG, MIT CTL analysis.

The problem: not all products behave the same, SC-wise

Supply Segments & Strategies | **Client Segments & Strategies**



versus

Agile SC for some products	Plan	Source	Make	Deliver	**Agile SC for some clients**
Efficient SC for some products	Plan	Source	Make	Deliver	**Efficient SC for some clients**
High SLA SC for some products	Plan	Source	Make	Deliver	**High SLA SC for some clients**

Source: adapted from GT Nexus: Supply Chain Segmentation Enabled, MIT CTL analysis.

We had one clear main objective

To support the **first step**
of Lenovo DCG's **customer-oriented** supply chain
by proposing **segmented policies** with analytics

Methodology: quantify portfolio to segment SC policies

Business Immersion, Data Collection	Feature Selection	Dimension Reduction (EFA)	k-Means Clustering	Workshop Discussion & Validation	Policies Framework Application	Machine Learning Loop
Diagnostics Define objectives Define scope Data gathering	Variable selection to describe Lenovo’s portfolio	Group variables to allow quant analysis	Identification of segments within portfolio	Segments critique and validation	SC breakdown Policies discussion for each segment	Design of continuous loop for segmentation implementation

Source: adapted from GT Nexus: Supply Chain Segmentation Enabled, MIT CTL analysis.

From 360,000+ data points from sales records…

The **Hyperscale BU was the focus** of this analysis due to its rapid sales growth

2018 was the first “full year” after Hyperscale’s sales ramp-up period

A “product” is a unique **Material Description** from Hyperscale 2018 FY sales records
A “client” is a standardized description based on the **Client Name** from the sales records

There are **142 unique client-product** pairs on the selected dataset (111 unique products)

Sales dataset: 54 columns, 13 relevant variables identified

1. **Sales in dollars** — sum of “Extended Price Shipped USD” for each product
2. **Sales in units** — sum of “Delivery Quantity” for each product
3. **Demand uncertainty** — Coefficient of variability of monthly sales (USD)
4. **Average Order Size** — Average value (USD sales) per order line
5. **Order Frequency** — Number of orders per year
6. **Distribution Channel** — Weighted average (by %USD sales) of distribution channel code
7. **Manufacturing LT** — Average days from order release to ship
8. **Shipping LT** — days from Order Shipping Date until Proof Of Delivery
9. **Total LT** — Average days from Order Receival to Proof Of Delivery
10. **DOT%** — Overall delivery on time % per product
11. **Sources of Supply** — Count of unique Source Of Supply per product
12. **Destinations** — Count of unique Ship to Country per product
13. **SC footprint (routes)** — Count of unique pair of Source of Supply and Destination per product

Variables: relevant descriptive features & target metrics

1 Sales in dollars
sum of “Extended Price Shipped USD” for each product

2 Sales in units
sum of “Delivery Quantity” for each product

3 Demand uncertainty
Coefficient of variability of monthly sales (USD)

4 Average Order Size
Average value (USD sales) per order line

5 Order Frequency
Number of orders per year

6 Distribution Channel
Weighted average (by %USD sales) of distribution channel code

7 Manufacturing LT
Average days from order release to ship

8 Shipping LT
days from Order Shipping Date until Proof Of Delivery

9 Total LT
Average days from Order Receival to Proof Of Delivery

10 DOT%
Overall delivery on time % per product

11 Source of Supply
Count of unique Source Of Supply per product

12 Destinations
Count of unique Ship to Country per product

13 SC footprint (routes)
Count of unique pair of Source of Supply and Destination per product

Exploratory Factor Analysis[1] identified two dimensions

1 Sales in dollars
sum of "Extended Price Shipped USD" for each product

2 Sales in units
sum of "Delivery Quantity" for each product

3 Demand uncertainty
Coefficient of variability of monthly sales (USD)

4 Average Order Size
Average value (USD sales) per order line

5 Order Frequency
Number of orders per year

6 Distribution Channel
Weighted average (by %USD sales) of distribution channel code

7 Manufacturing LT
Average days from order release to ship

8 Shipping LT
days from Order Shipping Date until Proof Of Delivery

9 Total LT
Average days from Order Receival to Proof Of Delivery

10 DOT%
Overall delivery on time % per product

12 Destinations
Count of unique Ship to Country per product

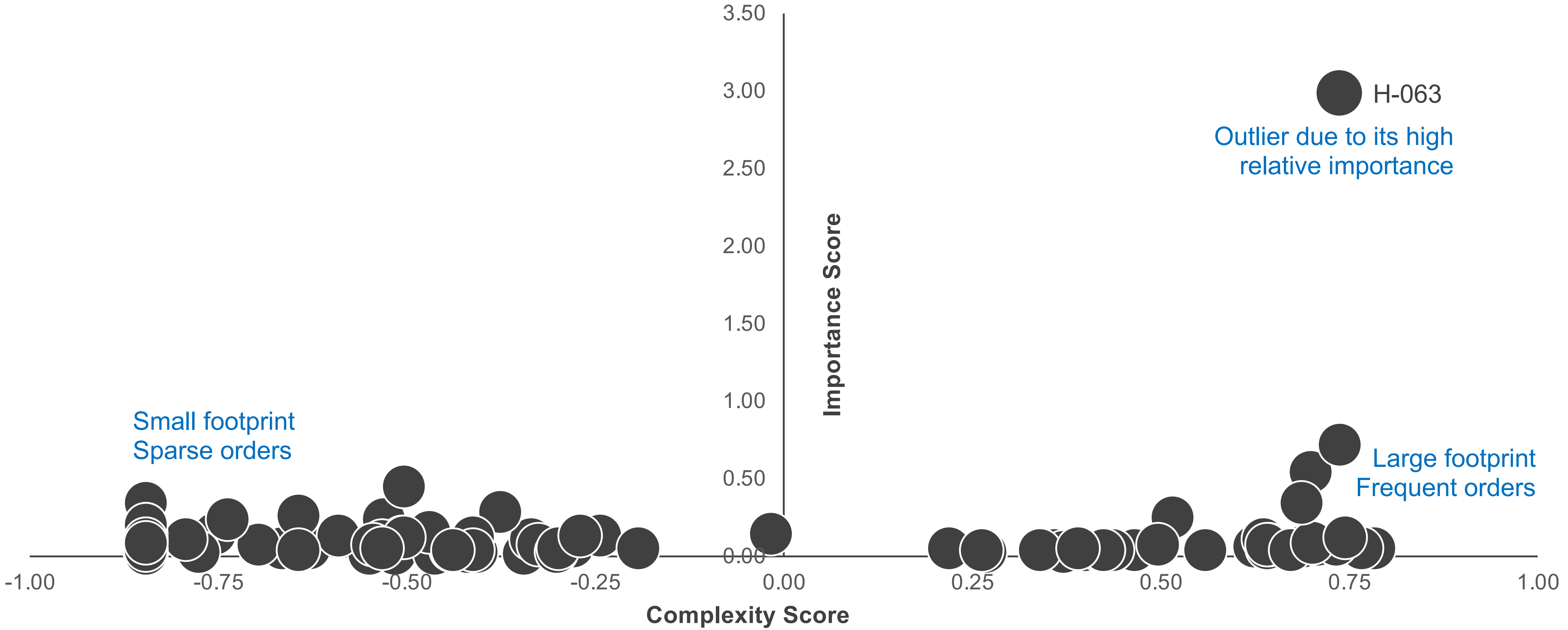
Targets

Lenovo

Source: MIT CTL. [1] Exploratory Factor Analysis (EFA) is a mathematical approach that reduces dimensions in the dataset in order to increase data analysis capacity. It groups variables (called *features* in the algorithm) into dimensions (called *factors* in the algorithm) that are able to explain the variance in the data with minimal data loss in the compression process.

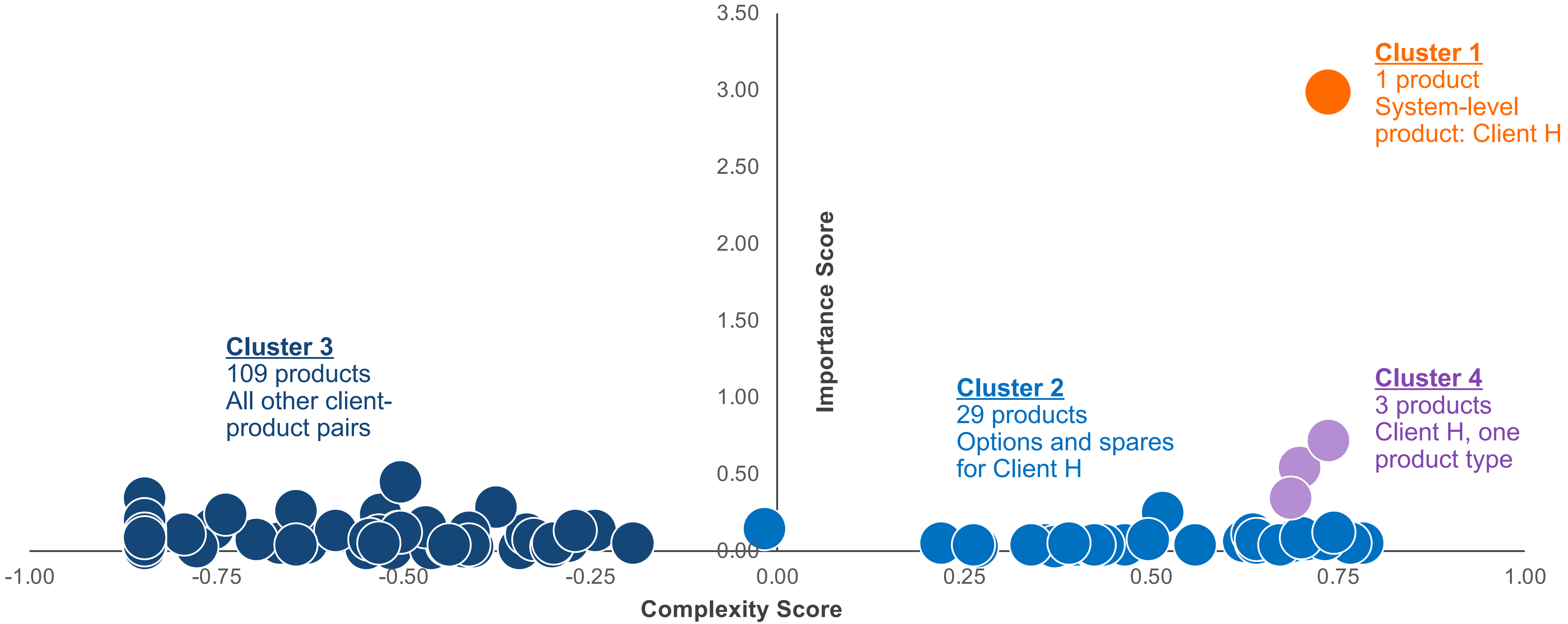
Each Client-Product is now described by its 2 dimensions

Complexity x Importance
(D1 x D2)



k-Means clustering explored 2 to 12 clusters[1], 4 was ideal

Portfolio Distribution on the Complexity x Importance Space
(D1 x D2, k=4)



Source: MIT CTL. [1]K is a user input for the K-Means model (hyperparameter). The more clusters, the more precise the analysis gets. However, we lose analytical power if K is too big. After defining a reasonable range (2 to 12) for iterations, methods can be applied to observe the results' accuracy, ex. Elbow and Silhouette Method. Final number of clusters is often based on those methods and is ultimately a user decision, not a model output.

Additional clusters simply divided C3 into smaller sets

Portfolio Clusters Overview for Multiple K
(D1 x D2, k=2 to 12, table is non-exhaustive)

# Clusters	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6
2	**1 product** Client H Lenovo HyperScale Node	**141 products** All other Client H All others Products	---	---	---	---
3	**1 product** Client H, Lenovo HyperScale Node	**32 products** Client H, all other products	**109 products** All other Clients	---	---	---
4	**1 product** Client H, Lenovo HyperScale Node	**29 products** Client H components & spare parts	**109 products** All other Clients	**3 products** Rack and Chassis	---	---
5	**1 product** Client H, Lenovo HyperScale Node	**28 products** Client H components & spare parts	**79 products** All other products for Clients H (i.e. ServerTS), misc	**3 products** Rack and Chassis	**31 products** All others, misc	---
6	**1 product** Client H, Lenovo HyperScale Node	**28 products** Client H components & spare parts	**78 products** All other products for Clients H (i.e. ServerTS), misc	**3 products** Rack and Chassis	**27 products** All others, misc	**5 products** Miscellaneous

(…)

Lenovo Source: MIT CTL.

Final recommended clusters are all statistically significant

Complexity and Importance among clusters
(Targets are not indexed averages: days, days, percentage of deliveries)

Clusters	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Cluster Description & Size	*System-level product (1)*	*Options and spares for Client H (29)*	*All other clients (109)*	*Client H, one product type (3)*
Importance Position	High	Low	Low	Median
Complexity Position	High	Median	Low	High
Target 1 | Total Lead Time	47.44	30.94	26.61	46.79
Target 2 | Mftg Lead Time	23.37	11.46	9.96	22.78
Target 3 | Delivery on Time	88%	75%	76%	89%

Lenovo

Source: MIT CTL.

Policies were then discussed and weighted for each cluster

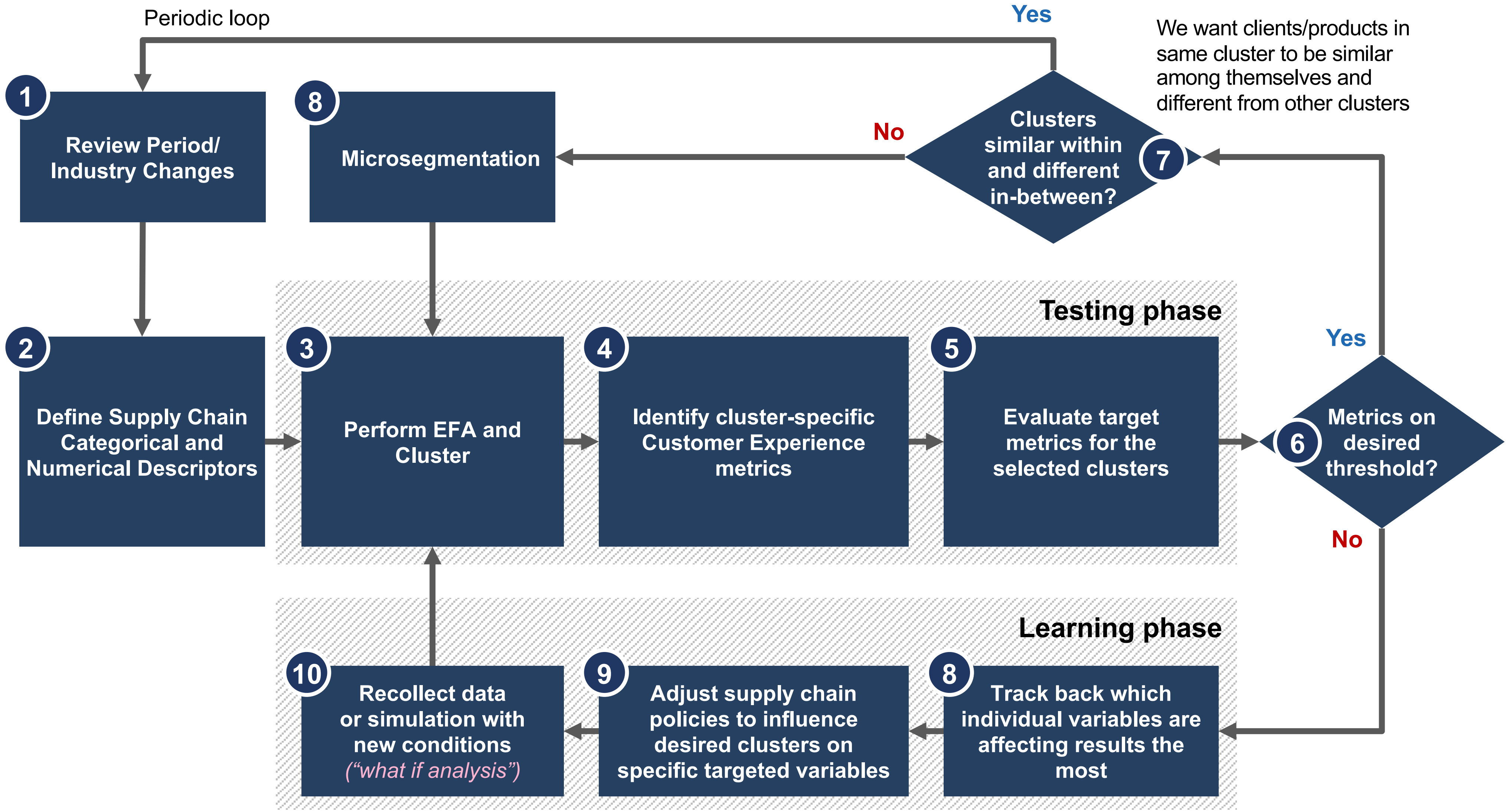
Policy-cluster matrix with importance weights from workshop

Relatively high importance — Relatively low importance

Clusters	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Cluster Description & Size	*System-level product (1)*	*Options and spares for Client H (29)*	*All other clients (109)*	*Client H, one product type (3)*
Sourcing				
Inventory				
Production			**Heterogeneous** Various clients Further segmentation to better understand cluster.	
Fulfillment				
Customers				

Source: MIT CTL, Lenovo DCG.

And a learning loop was kicked-off at Lenovo DCG



Value contribution to Lenovo DCG is clear and ongoing

1

Segmented Policy Design

Each cluster is mapped for distinct policies and supply chain requirements.

Machine Learning approach ensures continuous value addition.

2

Portfolio Management

Lenovo DCG is achieving high service levels across its portfolio without similar cost efficiency.

Potential short-term target: significant inventory reduction.

3

Data Management

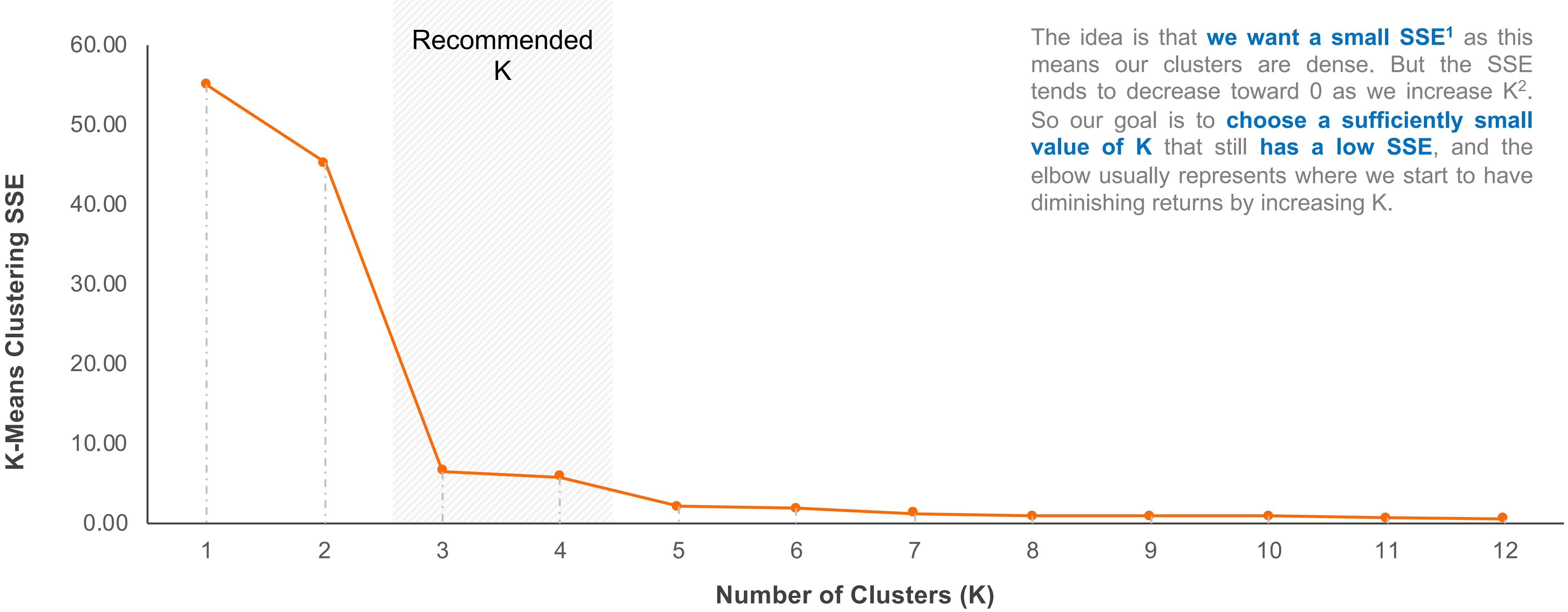
Identification of additional data for model improvement

Reruns are simple and easy with the ML approach

thanks.

Elbow recommends k=3 or 4, yet 4 enables better analysis

K-Means Clustering SSE (Sum of Square Errors[1]) x Number of Clusters for the Dataset
(Index, K)



The idea is that **we want a small SSE[1]** as this means our clusters are dense. But the SSE tends to decrease toward 0 as we increase K[2]. So our goal is to **choose a sufficiently small value of K** that still **has a low SSE**, and the elbow usually represents where we start to have diminishing returns by increasing K.

Source: MIT CTL. [1]The Sum of Square Errors (SSE) is a KPI that indicates the distance between each cluster centroid and all the data points in that cluster. [2] The SSE is 0 when k is equal to the number of data points in the dataset, because then each data point is its own cluster, and there is no error between it and the center of its cluster.