
CUBE
How to prioritize your product backlog to maximize your chances for success
Patrick Frey

About

- **Patrick Frey**
- **Passionate** about **product management**, **UX**, **agile development methods** & **value-based pricing**
- **> 12 years of experience** as **product manager** @ ETAS GmbH (100% subsidiary of **Bosch**)
- **Initiated & led an innovative software-based solution** (www.etas.com/ehandbook) **from idea to profitable business** used **at all major automotive OEMs & Tier-1's** worldwide
- **EHANDBOOK** won 3rd place in **"Product of the Year 2016"** competition by Automotive Electronics
- **Created CUBE method** based on **practical experiences** of creating a successful B2B software business with **200 sprints** & **30 product releases**

Contact

Email patrickfreyleanpm@gmail.com

LinkedIn https://www.linkedin.com/in/freypatrick

Contents

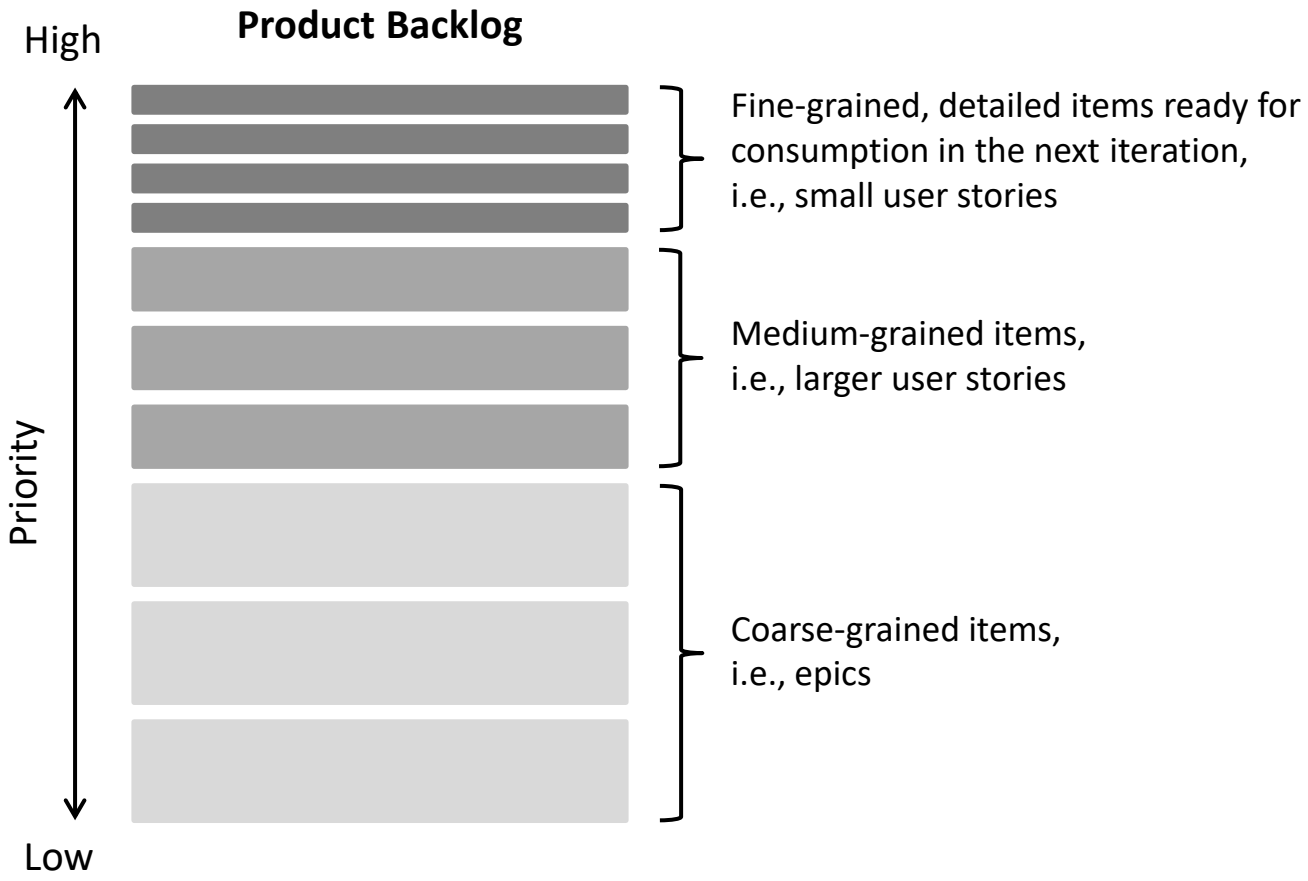
- Who can benefit from **CUBE method**?
- **Goals and constraints** for prioritization
- **Examining and prioritizing product backlog items** from different perspectives
  - The **Customer Perspective**
  - The **Business Perspective**
  - The **Development Perspective**
- **Bringing all Perspectives together**
- **Summary**
- **Disclaimer**

Who can benefit from CUBE method?

- In **agile methods** (e.g., Scrum, Kanban), the product backlog is central information repository containing work items
- Product backlogs are
  - **used for collaboration** between product owners and development teams
  - **basis for all kinds of planning activities** (e.g., sprint planning, release planning, roadmap planning)
- **Product owner (PO)** is responsible for maintaining the product backlog

➔ **The CUBE method** is **a tool for POs** to obtain a **reasonable prioritized product backlog.**

The Product Backlog



Source: Roman Pichler: Agile Product Management with Scrum

The Product Backlog

- A **product backlog** is a **prioritized list of work** items
  - **Items at top** have **high priority** („ready for work“, sufficiently detailed, estimated, etc.)
  - **Items at bottom** have **low priority** (i.e., coarse grain, no or not all details available, only roughly estimated, etc.)

The challenge of prioritizing

Prioritizing = **coming up** with an **order** in that work items from the backlog should **be taken up**

- **High priority** = take up **first**
- **Low priority** = take up **later / last**

The challenge of prioritizing

Common challenges:

- How to prioritize?
- What should come first, second, third? Why?
- What‘s the motivation and reasoning behind prioritization decisions?
- What are the objectives behind the prioritization?
- How to explain (or even defend) a specific prioritization?

➔ **The CUBE method is a systematic approach to derive a prioritized product backlog**

Where does CUBE originate from?

- In **2011**, I had the opportunity to start and then lead the development activities for a **new software-based solution called EHANDBOOK at ETAS GmbH**
- After 1 year of **user research**, a large list of „requirements“ was gathered (Excel)
  - Prioritization by domain experts at customers resulted in 40 items with prio #1, 30 with prio #2 and 10 with prio #3
  - Prioritization was not helpful for sequencing development efforts
- As development team, we had adopted **Scrum** (first team at ETAS) and wanted to **incrementally build a solution**.

Where does CUBE originate from?

- When searching for a **suitable method to analyze the customer and user requirements**, I came across the **KANO model.**
- **Development team** was using **planning poker** with **user story points** to estimate development effort
- As **product manager/product owner**, I sought for a means to express and compare **business value**. I discovered Business Value Game.

➔ **Combining the different methods led to creation of CUBE.**

Example backlog

- Small backlog for generic example of an online auction platform
- Used to demonstrate and explain CUBE concepts

Online auction platform
(e.g., eBay, ETSY)

ID	Description
1	Web version and Smarphone app
2	Allow unlimited pictures to upload
3	Automatic quality improvement of uploaded pictures
4	Provide user-defined product categories
5	Limit number of visitors per article

The CUBE method

The **CUBE prioritization** method is based on examining work items from three different perspectives

1. the **CUstomer perspective**
2. the **Business perspective**
3. the **development Effort perspective**

- The letters refer to the **perspective**
- **Order of the abbreviations** refers to **recommended order of performing examinations** – first CU, then B, then E

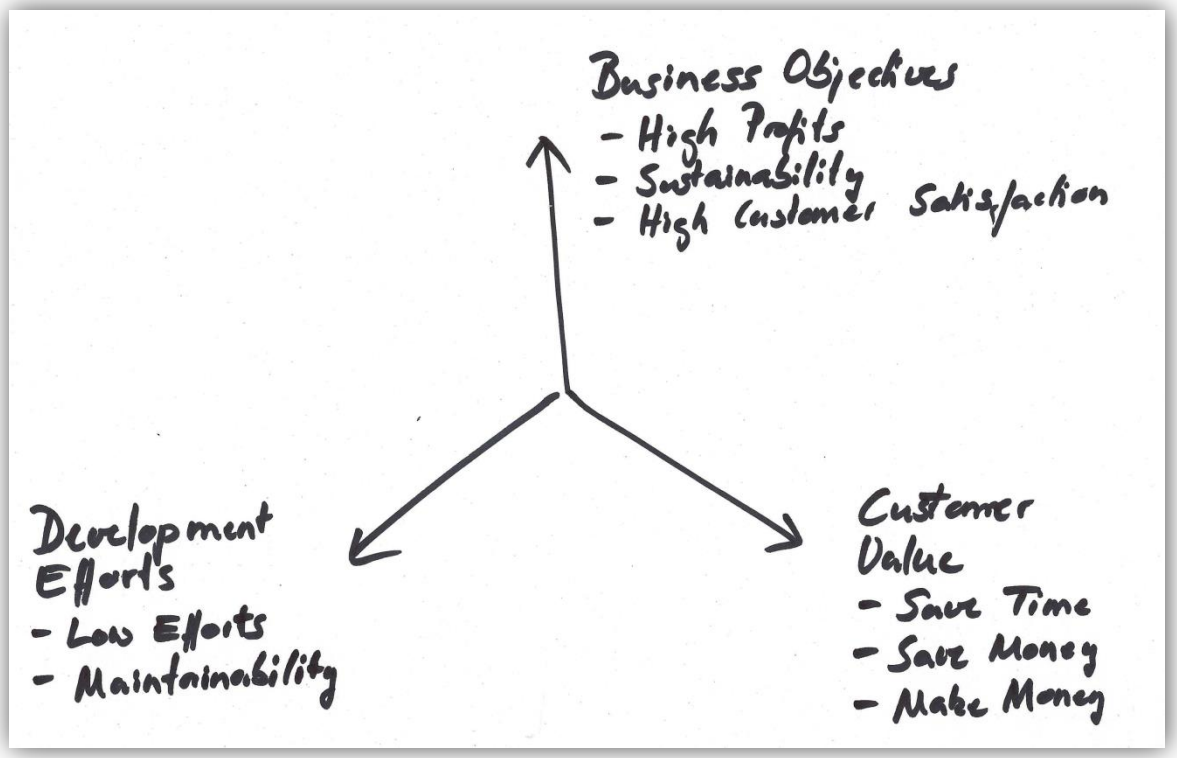
Goals and Constraints for Prioritization

Prioritization goals:

1. maximization of **customer** value
2. maximization of **business** value
3. minimization of **development** efforts

Goals and Constraints for Prioritization

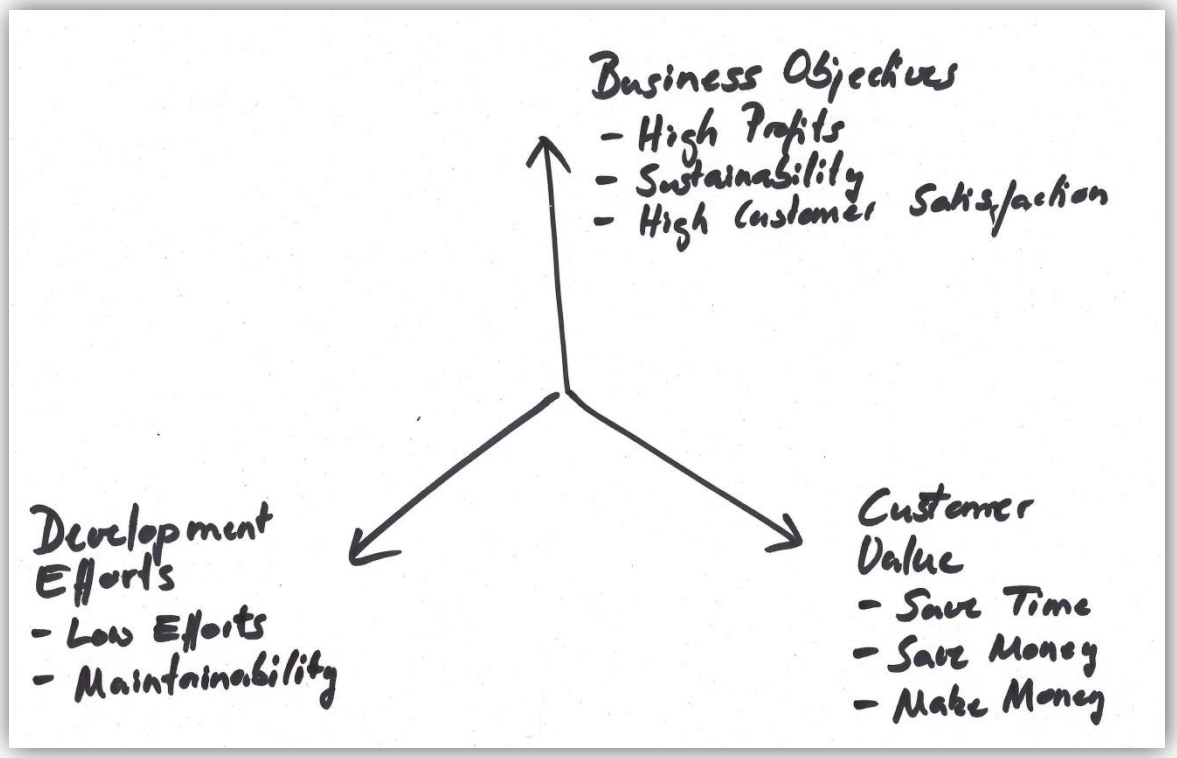
Challenge: Prioritization goals are not automatically aligned. They can pull in different directions!



Goals and Constraints for Prioritization

Effect of non-aligned prioritization goals:

Slow-down of progress ➔ threat to success.



Goals and Constraints for Prioritization

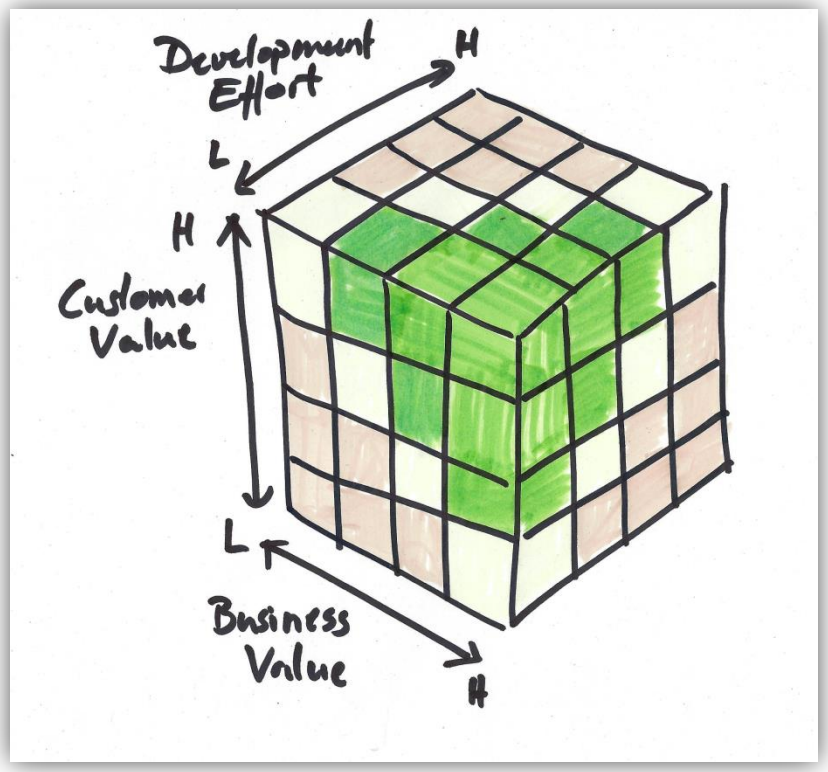
Constraints:

1. **technical** dependencies
2. **external** dependencies

Prioritization is an optimization problem

„Prioritizing“ is product owners game of **solving a Rubik‘s cube**

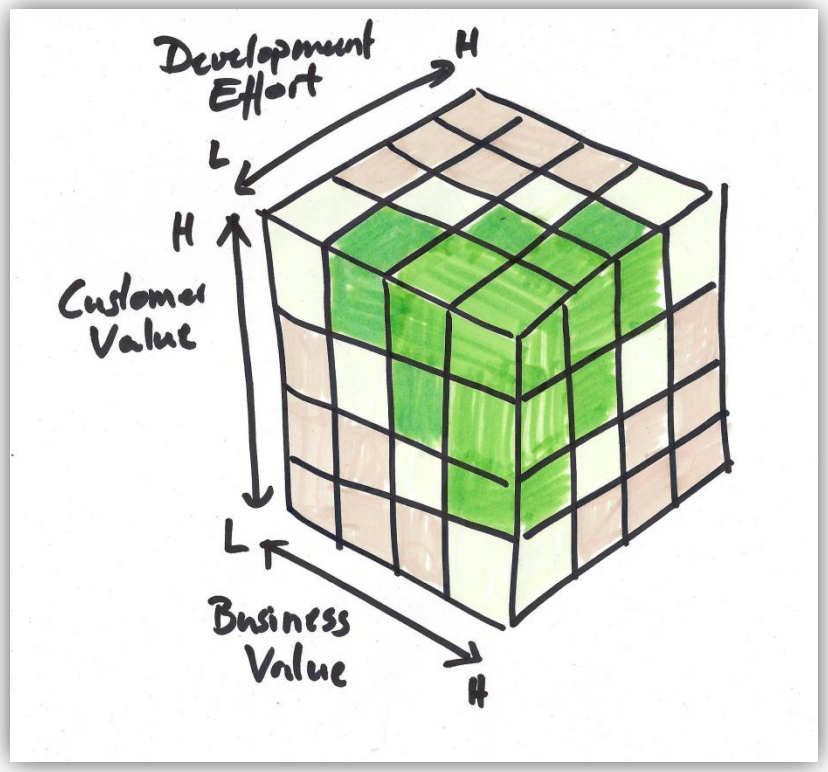
- Requires to **inspect items from different perspectives** to check how things fit together
- Requires **work** (turning / shifting / moving items)
- **Prioritization problem** is **„solved“** when issues (=colors) **„align“**



Prioritization is an optimization problem

CUBE helps to prioritize items with

1. High **customer** value
2. High **business** value
3. Low **development** effort

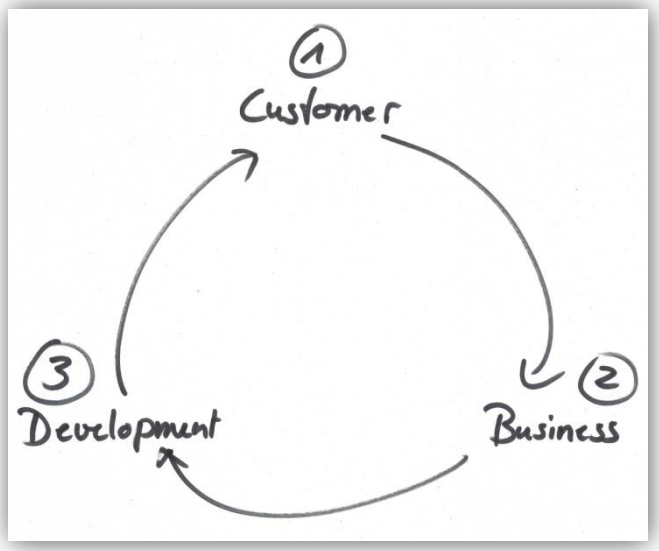


Examining and prioritizing product backlog items from different perspectives

1. The **customer perspective**:
   How would your customers / users be affected by shipping a specific backlog item?

2. The **business perspective**:
   How does realizing a backlog item affect your business?

3. The **development perspective**: How does working on a specific backlog item affects your development?



Examining and prioritizing product backlog items from different perspectives

Why start with the customer-perspective?

- Customer-centric organizations are more successful that others who are e.g. technology-centric (=key focus on development aspects)

> *The most important single thing is to focus obsessively on the customer. Our goal is to be earth's most customer-centric company.*
>
> Jeff Bezos, Amazon

➔ **Enforce customer-centricity by starting with customer value**

Examining and prioritizing product backlog items from different perspectives

Why follow-up with the business-perspective?

- Businesses need to make revenue to fund research & development activities, operations, etc.
- Projects typically do not run out of items to work on (e.g., derived customer needs), but money to fund the activities

➔ **Follow-up with business perspective to align customer and your own business interests**

Examining and prioritizing product backlog items from different perspectives

Why take development effort last?

- To address customer needs, there typically are **multiple solutions** with different levels of development effort
- **Engineers & managers** are typically **too idealistic** (wishful thinking; over-engineering / worst-case thinking)

➔ **Take development effort last to give "cheap" solutions a chance and not get lost in engineering rabbit-holes**

Categorizing backlog items and assigning abstract values

In CUBE, work items are **classified into categories** and **assigned with abstract estimation values**

Why?

- **Categorization** helps to **cluster items**
  - Overview
  - Taxonomy of backlog items
- **Abstract values** makes items **comparable** (relative size matters)
- **Categorization + abstract estimation values** help to **determine direction** and **maximize ROI** of the efforts spent

Categorizing backlog items and assigning abstract values

Customer Perspective	• **Categorization** of **customer value** according to **KANO model**
Business Perspective	• **Categorization** of **business impact** into **four Business Value Categories** • **Size estimated** in **Business Value points**
Development Perspective	• **Effort estimated** in **Story Points**

The Customer Perspective

Customer value is the most important key success factor for a new solution, product or service.

Typical challenges for **product teams***: Find out

- What is **important**?
- What is **not important**?
- **How important** is something **in relation to something else**?

*Product team consists at least of the roles product manager/owner, (lead) developer/architect and UX specialist

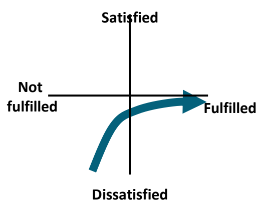
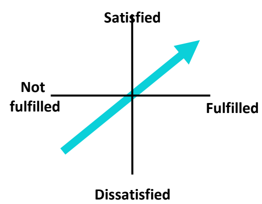
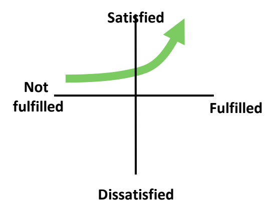
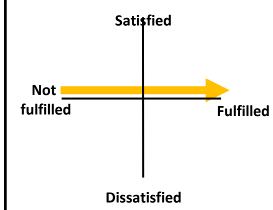
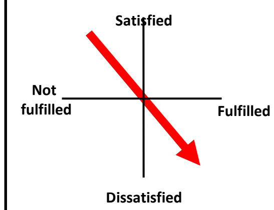
The KANO model

- Developed in the 1980s by **Noriaki Kano**
- A **theory** for **product development** and **customer satisfaction**
- **Classification** of **customer preferences** into **five categories**
  - Note: Different terminologies have been used for the categories (due to translation from Japanese to English)
  - For CUBE method, the terms introduced as follows are used

Noriaki Kano

Source:
https://www.uxness.in/2015/07/kano-model.html

The KANO model

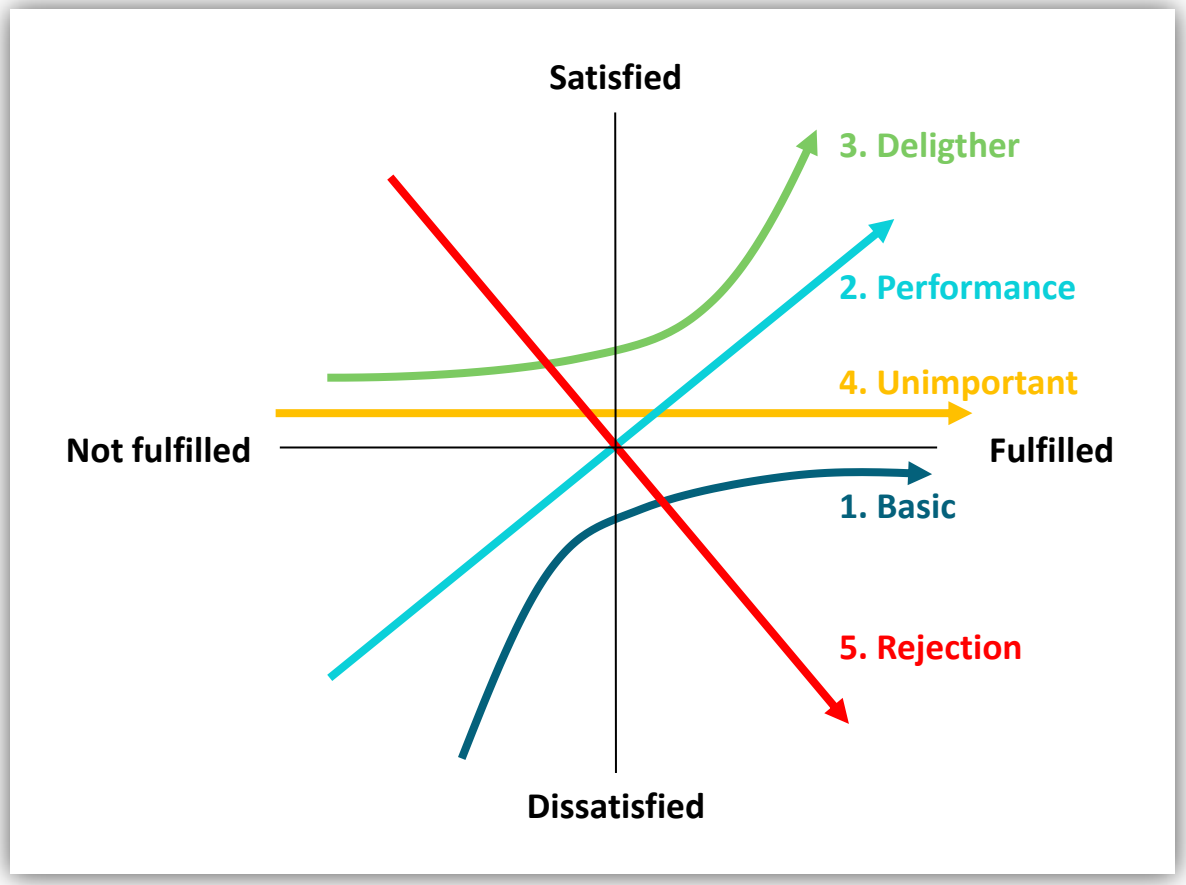
Category	Basic	Performance	Delighter	Unimportant	Rejection
User perspective (Classification from KANO model)	Implicit; taken for granted User is dissatisfied if not fulfilled	User is aware; the more, the merrier Creates customer value - removes dissatisfaction	Unexpected; positive surprise Customer value beyond expectation	User is indifferent / doesn't care No or little customer value	Dissatisfaction if present Negative customer value
Degree of fulfillment of needs / expectations & effect on customer satisfaction					

The KANO model

KANO diagram:

Degree of fulfillment of customer needs / expectations &

effect on customer satisfaction



The KANO model

Category	Basic	Performance	Delighter	Unimportant	Rejection
User perspective (Classification from KANO model)	Implicit; taken for granted User is dissatisfied if not fulfilled	User is aware; the more, the merrier Creates customer value - removes dissatisfaction	Unexpected; positive surprise Customer value beyond expectation	User is indifferent / doesn't care No or little customer value	Dissatisfaction if present Negative customer value
Business perspective (Interpretation of KANO model classification)	No differentiation from competitors	(Some) differentiation from competitors	Can be strong differentiation from competitors Unique selling point	Unimportant for differentation from competitors	To be avoided

The KANO model

Category	Basic	Performance	Delighter	Unimportant	Rejection
Example Hotel	Clean room	Early checkin Late checkout Free Wifi	Free parking Free entrance to special tourist attraction Skiing pass included	Room equipped with radio	No pets allowed (if travelling with pets) Only cold water in shower No TV

Applying the KANO model to your product backlog

- Transfer backlog items to a **spreadsheet** (Excel or Google Sheets)
- Add column **„Customer Value“**
- For each item, **select at least one category**
  - Basic
  - Performance
  - Deligther
  - Unimportant
  - Rejection

Where do categorizations come from?

1. From **user research**
2. Applying **domain expertise**
3. Taking an **educated guess**

Conducting user research is **strongly recommended!**

Applying the KANO model to your product backlog

- Transfer backlog items to a **spreadsheet** (Excel or Google Sheets)
- Add column **„Customer Value“**
- For each item, **select at least one category**
  - Basic
  - Performance
  - Deligther
  - Unimportant
  - Rejection

Online auction platform
(e.g., eBay, ETSY)

ID	Description	Customer Value
1	Web version and Smarphone app available	Basic
2	Allow unlimited pictures to upload	Performance
3	Automatic quality improvement of uploaded pictures	Deligther
4	Provide user-defined product categories	Unimportant
5	Limit number of visitors per article	Rejection

Conducting user research to support the KANO model classification

KANO model provides a very **systematic approach** for **user interviews**

Ask the following **two questions**:

1. What would you say **if the product / solution could do / would have** <feature/capability XYZ>? ⇨ **Functional** needs
2. What would you say **if the product / solution could not do / would not have** <feature/capability XYZ>? ⇨ **Disfunctional** needs

Conducting user research to support the KANO model classification

Only the **following possible answers** are **allowed**:

1. I would be very happy
2. I take this as granted
3. I don’t care
4. I would barely accept this
5. I would be very annoyed

Conducting user research to support the KANO model classification

	Functional	**Disfunctional**
Template	What would you say if the product / solution **could do / would do … ?**	What would you say if the product / solution **could *not* do / would *not* do …?**
I would be very happy		
I take this for granted		
I don't care		
I would barely accept this		
I would be very annoyed		

Conducting user research to support the KANO model classification

	Functional	Disfunctional
Answers for Basic Needs	What would you say if the product / solution **could do / would do … ?**	What would you say if the product / solution **could *not* do / would *not* do …?**
I would be very happy	**X**	
I take this for granted	**X**	
I don't care		
I would barely accept this		
I would be very annoyed		**X**

Conducting user research to support the KANO model classification

	Functional	Disfunctional
Answers for Performance Features	What would you say if the product / solution **could do / would do … ?**	What would you say if the product / solution **could *not* do / would *not* do …?**
I would be very happy	X	
I take this for granted		
I don't care		
I would barely accept this		X
I would be very annoyed		

Conducting user research to support the KANO model classification

	Functional	Disfunctional
Answers for Delighter Capabilities	What would you say if the product / solution **could do / would do … ?**	What would you say if the product / solution **could *not* do / would *not* do …?**
I would be very happy	X	
I take this for granted		
I don't care		X
I would barely accept this		
I would be very annoyed		

Conducting user research to support the KANO model classification

	Functional	Disfunctional
Answers for Unimportant Features	What would you say if the product / solution **could do / would do … ?**	What would you say if the product / solution **could *not* do / would *not* do …?**
I would be very happy		
I take this for granted		
I don't care	X	X
I would barely accept this		
I would be very annoyed		

Conducting user research to support the KANO model classification

	Functional	Disfunctional
Answers for Rejection Points	What would you say if the product / solution **could do / would do … ?**	What would you say if the product / solution **could *not* do / would *not* do …?**
I would be very happy		**X**
I take this for granted		**X**
I don't care		
I would barely accept this		
I would be very annoyed	**X**	

Customer value categories can change over time

Over time, **capabilities / features that delight** will eventually **become performance requirements** and **later unarticulated basic needs**.

„Free WiFi in all areas of the hotel“	**Delighter**	**Performance**	**Basic**	**Basic**
Functional answer	„I would be very happy“	„I would be very happy“	„I take this for granted“	„I take this for granted“
Disfunctional answer	„I don’t care“	„I would barely accept this “	„I would be very annoyed “	„I would be very annoyed “

➔**Continuously seek product and service innovation!**

Customer value categories can change over time

Over time, **capabilities / features that delight** will eventually **become performance requirements** and **later unarticulated basic needs**.

	2005	2010	2015	2020 → Time
„No free WiFi in all areas of the hotel“	**Unimportant**	**Unimportant**	**Rejection**	**Rejection**
Functional answer	„I don’t care“	„I don’t care“	„I would be very annoyed “	„I would be very annoyed “
Disfunctional answer	„I don’t care“	„I don’t care“	“I would be very happy”	„I take this for granted“

➔**Continuously seek product and service innovation!**

The KANO model

The KANO Model helps to understand the main categories of customer requirements

- **Prioritize development efforts**
- **Identify gaps** in offerings

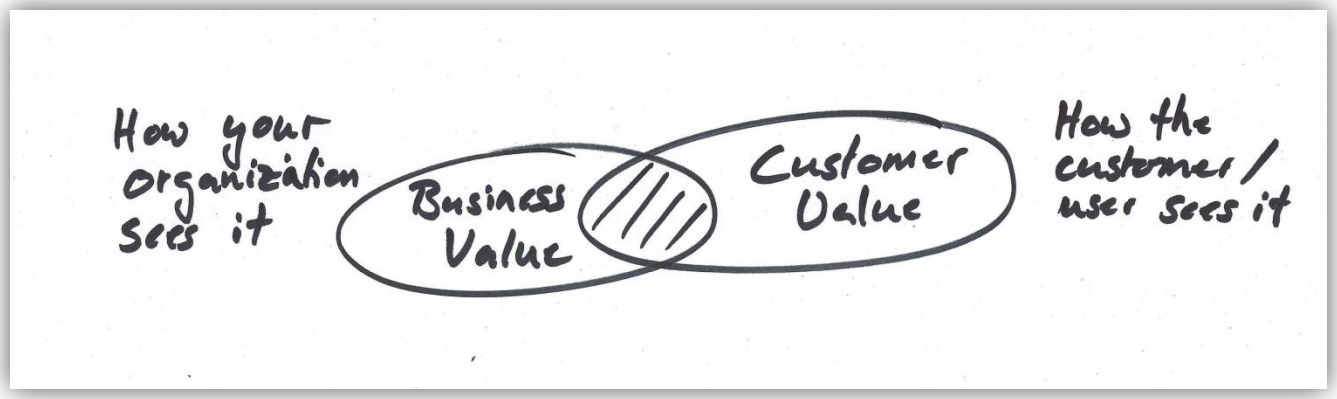
Depending on phase of your project / product, results from KANO Model can be used for different purposes:

- **Validate MVP / new concept:** Focus on **Delighters**
- **Improve Quality / UX:** Focus on removing **rejections**
- **Derive a roadmap of meaningful product increments:** Focus on **Basic** + some **Deligthers**; add **Performance** (as required)

The Business Perspective

- **Businesses** require **revenues** to **fund activities** such as research, development and operations.
- **Revenues** are **obtained** from **existing and new customers**.

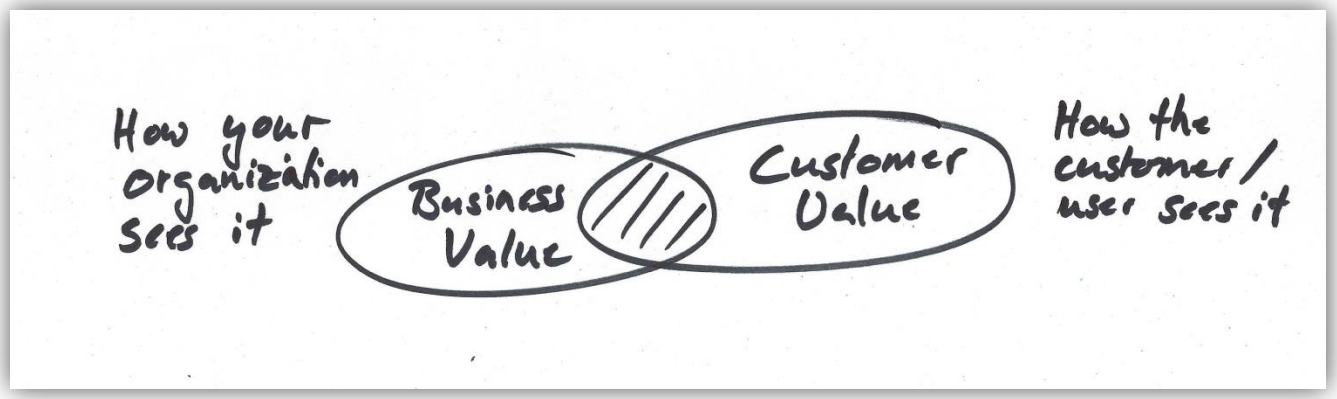
- Customer Values and Business Values are related, but not the same



The Business Perspective

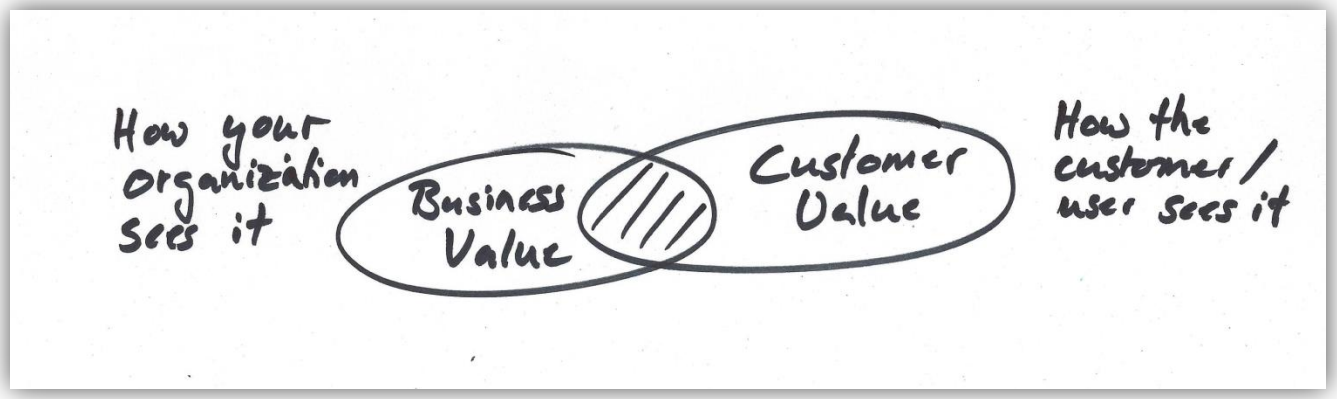
Customer Value and **Business Value** are **related**,

but are <u>not</u> the same



It‘s a **matter of perspective** how values are seen

The Business Perspective



Examples:

- A **free lunch** has **high customer value**, but **no** (or only indirect) **business value**
- **Acqusition fees** (e.g., for insurance contracts) have **high business value**, but **no** (direct) **customer value**

Business Value Categories

- **Customers** can typically be classified into two categories: **new customers** and **existing** customers
- Any new features / capabilities can help to **acquire new customers** or **serve existing customers**, or **both**
- New features / capabilities **can be sold separately** (upgrade) or **added free-of-cost to** the next version (update)
- **Some efforts** that you spent **do not directly serve your customer / user**, but help you to **operate your business** or **get better at it**

Business Value Categories

CUBE method uses **four business value categories**

New Business	Up Sell	Retainment	Operational Efficiency
Every feature that will potentially **bring new customers or new markets** will also bring a fresh flow of Money	Every feature that will potentially **bring money from existing customers** and could be sold as **add-on, upgrade or plug-in**	Every feature that will **avoid losing existing customers** will avoid the company losing money as well	Every feature that will allow the company to **save money (costs)** given a potential increase in any operation (installation, configuration, customization, …)

Relation between business value and customer value categories

Customer Value Category	Business Value Category	Rationale
Basic	New Business	Basic customer needs can be different from customer to customer (esp. in B2B). Anyways, they need to be fulfilled to win a new customer.
Performance	Retainment	Existing customers often demand better performance of existing features as they want to optimize they workflows.
Unimportant	Operational Efficiency	Many efforts that simply need to get done to develop, maintain or operate a product or service are not important to the user or customer. But they can be very valueable for your internal efforts.

Applying business value categories to your product backlog

- Transfer backlog items to a spreadsheet (Excel or Google Sheets)
- Add column **„Business Value Category“**
- For each item, select at least one category
  - New Business
  - Up-Sell
  - Retainment
  - Operational Efficiency

Online auction platform
(e.g., eBay, ETSY)

ID	Description	Business Value Category
1	Web version and Smarphone app available	New Business
2	Allow unlimited pictures to upload	Retainment Up-Sell
3	Automatic quality improvement of uploaded pictures	Retainment
4	Provide user-defined product categories	Operational Efficiency
5	Limit number of visitors per article	Operational Efficiency

Business Value Estimates

- Approach to assign business values to product backlog items is **inspired by Andrea Tomasini** (2007)
  - **Business Value Game** (akin to Planning Poker for estimating development efforts)
- **Business value estimates** are a **means to measure business value**
- In CUBE, the **scale** to express business value is **100, 200, 300, 500, 800, 1200, 2000, 3000**.
  - Inspired by **Fibonacci-series** (higher values are rounded)
  - **Multiplication factor 100** is later helpful to calculate a ration between development effort and business value
  - **Higher numbers indicate higher business value**

Business Value Estimates

Benefit of abstract numbers (i.e., without any unit)

- Allows for **relative comparison**
- *„X is better than Y because of …“*

How to come up with business value numbers?

1. **Product management board** (i.e., responsible PMs/Pos; representatives from Sales & Support) plays **„Business Value Game“**
2. Take an **educated guess**

Business Value Game

1. Item is presented and discussed (**initial clarification**).
2. Each participant **secretly estimates**.
3. All participants reveal their estimates.
4. **Group discusses** different estimates **and concludes on business value estimate** to be assigned

Business Value Game

Procedure:
If the Product Backlog is not yet estimated: The stakeholders select the smallest entry and assign the BV 200 to it.

For every feature (granularity: epic / theme): The Product Owner explains the theme / epic. Then:

1. Every participant selects (hidden!) one of his cards to estimate the relative Business Value.
2. All participants show the cards at the same time.
3. If estimations diverge, the two participants with the highest and lowest estimation explain their point of view. The procedure is repeated 2x at most.
4. If there is no agreement, then the smallest number is assigned.*

Timebox: No longer than 2h (others: 2-4h). If necessary, call for another meeting.

Applying business value estimates to your product backlog

- Add column **„Business Value Estimate“**
- Go through all items of one Business Value Category
  - Allows relative comparison
- Provide **Business Value Estimate** as number

Online auction platform
(e.g., eBay, ETSY)

ID	Description	Business Value Category	Business Value Estimate
1a	Smarphone app	New Business	3000
1b	Web version	New Business	2000
2	Allow unlimited pictures to upload	Retainment Up-Sell	1200
3	Automatic quality improvement of uploaded pictures	Retainment	800
4	Provide user-defined product categories	Operational Efficiency	800
5	Limit number of visitors per article	Operational Efficiency	500

The Development Perspective

- **Knowing the effort** it takes to deliver a backlog item is **crucial for planning activities**.

- **Story Points (SPs)** are widely used to express effort & complexity (=uncertainty).

- Story Points …
  - … can be applied to **Themes, Epics and User Stories**.
  - … are **abstract numbers** without a unit (no direct relation to hours, days, etc.)
  - … allow for **relative comparison** of backlog items concerning the effort

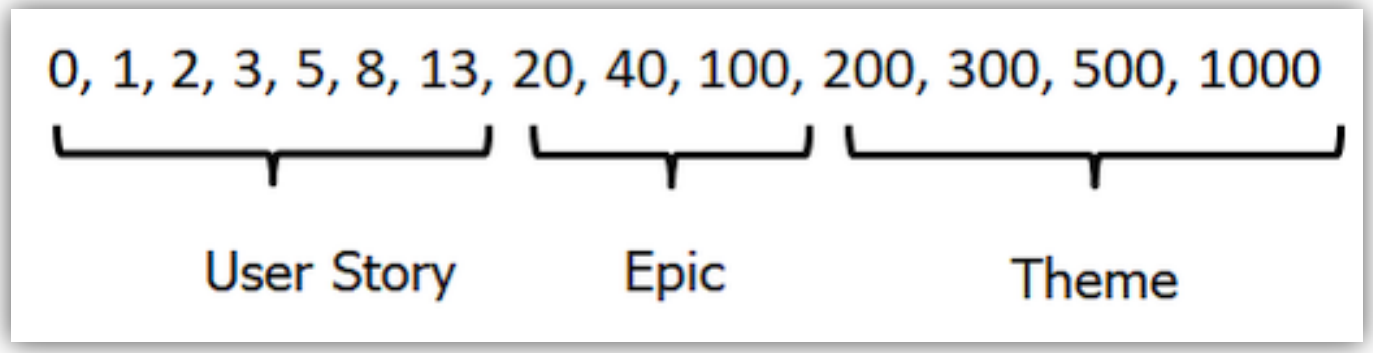
Common pitfalls with Story Points

- Story Points **cannot be used to compare the performance** of different teams.
  - **Each team has a different "calibration"** of their understanding of the size and complexity of a user story.

Estimating Development Efforts

- **Story Points** are expressed in **abstract numbers**
  - Inspired by the **Fibonacci** sequence
  - 0, 1, 2, 3, 5, 8, 13, 20, 40, 70, 100, 200, 500, 1000

- **Recommended interpretation** (for teams new to Scrum / using story points)



Estimating Development Efforts

1. **Product Owner explains** the Theme, Epic, User Story
2. A (hopefully) vital **discussion** arises
   - Team members **ask questions**
   - PO and other team members can (hopefully) clarify.
   - Typically, **senior developers** or **architects** can give some **guidance** how the user story can be realized. **Goal: drive out uncertainty** and **identify pending risks**.
3. All team members **estimate secretly**. The **estimates are revealed at once**.
4. **If numbers divert**, the team members with **highest** and **lowest** numbers **explain their rationale**. The team discusses and **agrees on a effort estimate**.

Tips for efficient collaboration between POs a developers

- Recommendation to POs: **do not question the effort estimate agreed by the team**. They are the experts, and they have to do the work. It is ***their*** effort estimate.
- If there is **uncertainty** expressed by the team, it's the **POs job to clarify**.
- You can **re-write user stories** to **flesh out uncertain parts**.
- **Work together**, not against each other. The **PO is part of the team**.

Applying effort estimates to your product backlog

- Add column **„Development Effort “** to your spreadsheet
- Go through items in a chosen order
  - New Business
  - Retainment
  - Operational Effiency
  - Up-Sell
- Provide an Effort Estimate

Online auction platform
(e.g., eBay, ETSY)

ID	Description	Develop-ment Effort	Rationale
1a	Smarphone app	100	Complexity to support iOS and Android
1b	Web version	40	Team knows what to do, but it‘s effort and takes some time
2	Allow unlimited pictures to upload	8	Just need to remove existing limiting factor
3	Automatic quality improvement of uploaded pictures	70	Need to learn AI and introduce new technology
4	Provide user-defined product categories	20	Larger new feature
5	Limit number of visitors per article	13	Smaller new feature

Bringing all perspectives together

- Each perspective for itself already adds a lot of insights
- Bringing all perspectives together helps to come up with meaningful prioritization that focuses on
  1. High customer value
  2. High business value
  3. Low development effort

Bringing all perspectives together

ID	Description	Customer Value	Bus. Value Category	Bus. Value Estimate	Dev. Effort
1a	Smarphone app	Basic	New Business	3000	100
1b	Web version	Basic	New Business	2000	40
2	Allow unlimited pictures to upload	Performance	Retainment Up-Sell	1200	8
3	Automatic quality improvement of uploaded pictures	Deligther	Retainment	800	70
4	Provide user-defined product categories	Unimportant	Operational Efficiency	800	20
5	Limit number of visitors per article	Rejection	Operational Efficiency	500	13

Backlog with details from all perspectives

- Provides transparency
- Basis for prioritization activities

Bringing all perspectives together

As both **business value** and **development effort** are estimated in **abstract numbers**, it allows us calculate a **"Return on Investment“ (ROI)**

$$ROI = \frac{Business\ Value}{Development\ Effort}$$

ROI expresses how much business value is created for investing 1 story point of development effort

Bringing all perspectives together

$$ROI = \frac{Bus.Value}{Dev.Effort}$$

ID	Description	Customer Value	Bus. Value Category	Bus. Value Estimate	Dev. Effort	ROI
1a	Smarphone app	Basic	New Business	3000	100	**30**
1b	Web version	Basic	New Business	2000	40	**50**
2	Allow unlimited pictures to upload	Performance	Retainment Up-Sell	1200	8	**30**
3	Automatic quality improvement of uploaded pictures	Deligther	Retainment	800	70	**11,5**
4	Provide user-defined product categories	Unimportant	Operational Efficiency	800	20	**40**
5	Limit number of visitors per article	Rejection	Operational Efficiency	500	13	**38,5**

Bringing all perspectives together

ID	Description	Customer Value	Bus. Value Category	Bus. Value Estimate	Dev. Effort	ROI
1a	Smarphone app	**Basic**	**New Business**	3000	100	**30**
1b	Web version	**Basic**	**New Business**	2000	40	**50**
2	Allow unlimited pictures to upload	Performance	Retainment Up-Sell	1200	8	**30**
3	Automatic quality improvement of uploaded pictures	Deligther	Retainment	800	70	**11,5**
4	Provide user-defined product categories	Unimportant	Operational Efficiency	800	20	**40**
5	Limit number of visitors per article	Rejection	Operational Efficiency	500	13	**38,5**

Insight #1:
It is more efficient to start with web-version although smartphone app has higher business value

Bringing all perspectives together

ID	Description	Customer Value	Bus. Value Category	Bus. Value Estimate	Dev. Effort	ROI
1a	Smarphone app	**Basic**	**New Business**	3000	100	**30**
1b	Web version	**Basic**	**New Business**	2000	40	**50**
2	Allow unlimited pictures to upload	Performance	Retainment Up-Sell	1200	8	**30**
3	Automatic quality improvement of uploaded pictures	Deligther	Retainment	800	70	**11,5**
4	Provide user-defined product categories	Unimportant	Operational Efficiency	800	20	**40**
5	Limit number of visitors per article	Rejection	Operational Efficiency	500	13	**38,5**

Insight #2:
Although there is a high ROI, realizing this item is waste due to customer rejection

Bringing all perspectives together

ID	Description	Customer Value	Bus. Value Category	Bus. Value Estimate	Dev. Effort	ROI
1a	Smarphone app	Basic	New Business	3000	100	**30**
1b	Web version	Basic	New Business	2000	40	**50**
2	Allow unlimited pictures to upload	**Performance**	**Retainment Up-Sell**	1200	8	**30**
3	Automatic quality improvement of uploaded pictures	Performance Deligther	Retainment	800	70	**11,5**
4	Provide user-defined product categories	Unimportant	Operational Efficiency	800	20	**40**
5	Limit number of visitors per article	Rejection	Operational Efficiency	500	13	**38,5**

Insight #3:
Clear indication which performance feature to take up first

Creating packages of product backlog items for incremental releases

Agile methods allow for **incremental product development**

- **Regular** or **coninuous releases**
  - **Regular = end of each month** or **quarter**
  - **Continuous =** whenever something is **„ready to ship“**

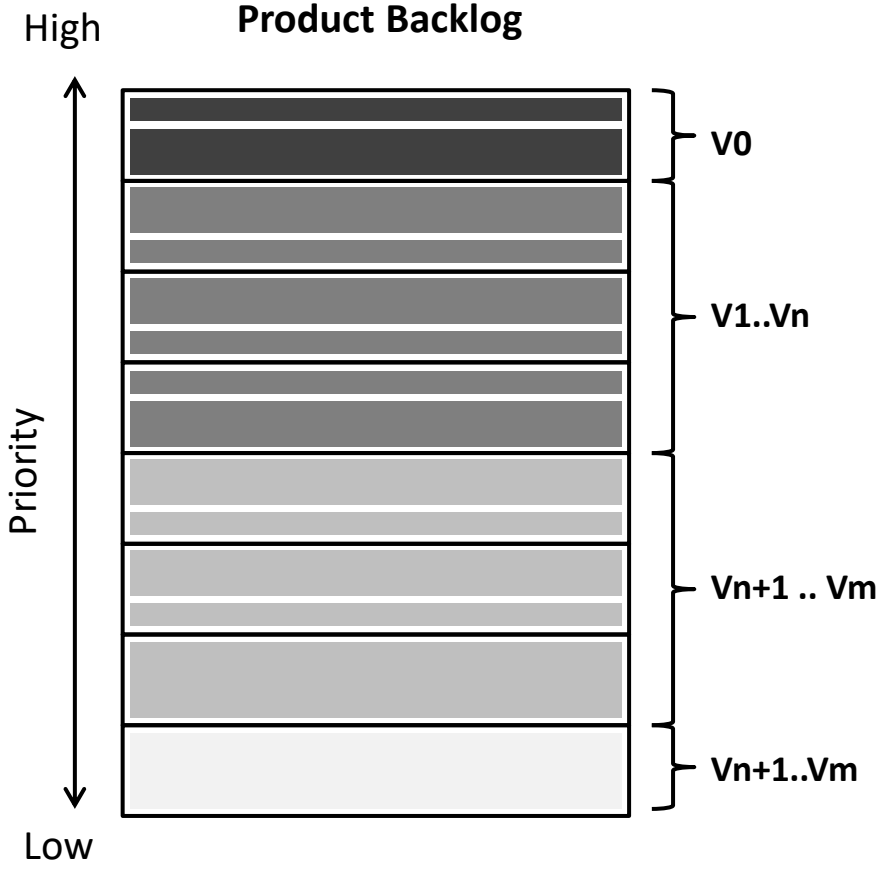
Planning of **incremental releases** requires to identify and define **consistent product scopes / packages**

Creating packages of product backlog items for incremental releases

Version	Goal	Criteria for selecting issues from product backlog
V0	Setting up the infrastructure	Everything that is required to set things up (e.g., the development infrastructure for developing, testing and shipping a new product).
V1 .. Vn	Initial versions („MVP“)	Focus on issues that provide "Basic" customer value and some “Performance” and “Deligthers” to address "New Business“.
Vn+1 .. Vm	Expanded versions for repeatable business	Add further issues that provide additional customer value in area of "Performance" / "Deligthers" for "Retainment“.
Vm+1	Maintain and expand established product / service	Focus on "Operational Efficiency" & “Up Sell” ➔ Automation; Quality; Broaden Customer / User Base; Scaling

Achieve **Product-Market-Fit** asap

Creating packages of product backlog items for incremental releases



Setting up the infrastructure
Everything that is required to set things up

Initial versions („MVP“)
Focus on issues that provide "Basic" customer value and some “Performance” and “Deligthers” to address "New Business“.

Expanded versions for repeatable business
Add further issues that provide additional customer value in area of "Performance" / "Deligthers" for "Retainment“.

Maintain and expand established product / service
Focus on "Operational Efficiency" & “Up Sell”

Bringing all perspectives together

ID	Description	Customer Value	Bus. Value Category	Bus. Value Estimate	Dev. Effort	ROI	Version
1a	Smarphone app	Basic	New Business	3000	100	**30**	**V3**
1b	Web version	Basic	New Business	2000	40	**50**	**V1**
2	Allow unlimited pictures to upload	Performance	Retainment Up-Sell	1200	8	**30**	**V2**
3	Automatic quality improvement of uploaded pictures	Performance Deligther	Retainment	800	70	**11,5**	**V4**
4	Provide user-defined product categories	Unimportant	Operational Efficiency	800	20	**40**	**V5**
5	Limit number of visitors per article	Rejection	Operational Efficiency	500	13	**38,5**	**Don‘t do**

Summary (1/2)

- **Prioritizing work items** of a product backlog **requires substantial efforts** and is **not an easy task**.
- The **CUBE method** can greatly **help** you to **take more rationale prioritization decisions** by taking **different perspectives** into account. This helps you to **maximize your odds of success**.
- **Goals** that should be reflected in the prioritization are the **maximization of customer value**, the **maximization of business value** and the **minimization of development efforts**.

Summary (2/2)

- There exist **constraints** such as **technical** or **external dependencies** that also need to be taken into account.
- **Examining** product backlog items **from different perspectives** creates transparency and insights for **taking better prioritization decisions**.
- The three most relevant perspectives are the **customer perspective**, the **development perspective** and the **business perspective**.

About

- **Patrick Frey**
- **Passionate** about **product management**, **UX**, **agile development methods** & **value-based pricing**
- **> 12 years of experience** as **product manager** @ ETAS GmbH (100% subsidiary of **Bosch**)
- **Initiated & led an innovative software-based solution** (www.etas.com/ehandbook) **from idea to profitable business** used **at all major automotive OEMs & Tier-1's** worldwide
- **EHANDBOOK** won 3rd place in **"Product of the Year 2016"** competition by Automotive Electronics
- **Created CUBE method** based on **practical experiences** of creating a successful B2B software business with **200 sprints** & **30 product releases**

Contact

Email patrickfreyleanpm@gmail.com

Twitter @patrickcfrey

LinkedIn https://www.linkedin.com/in/freypatrick