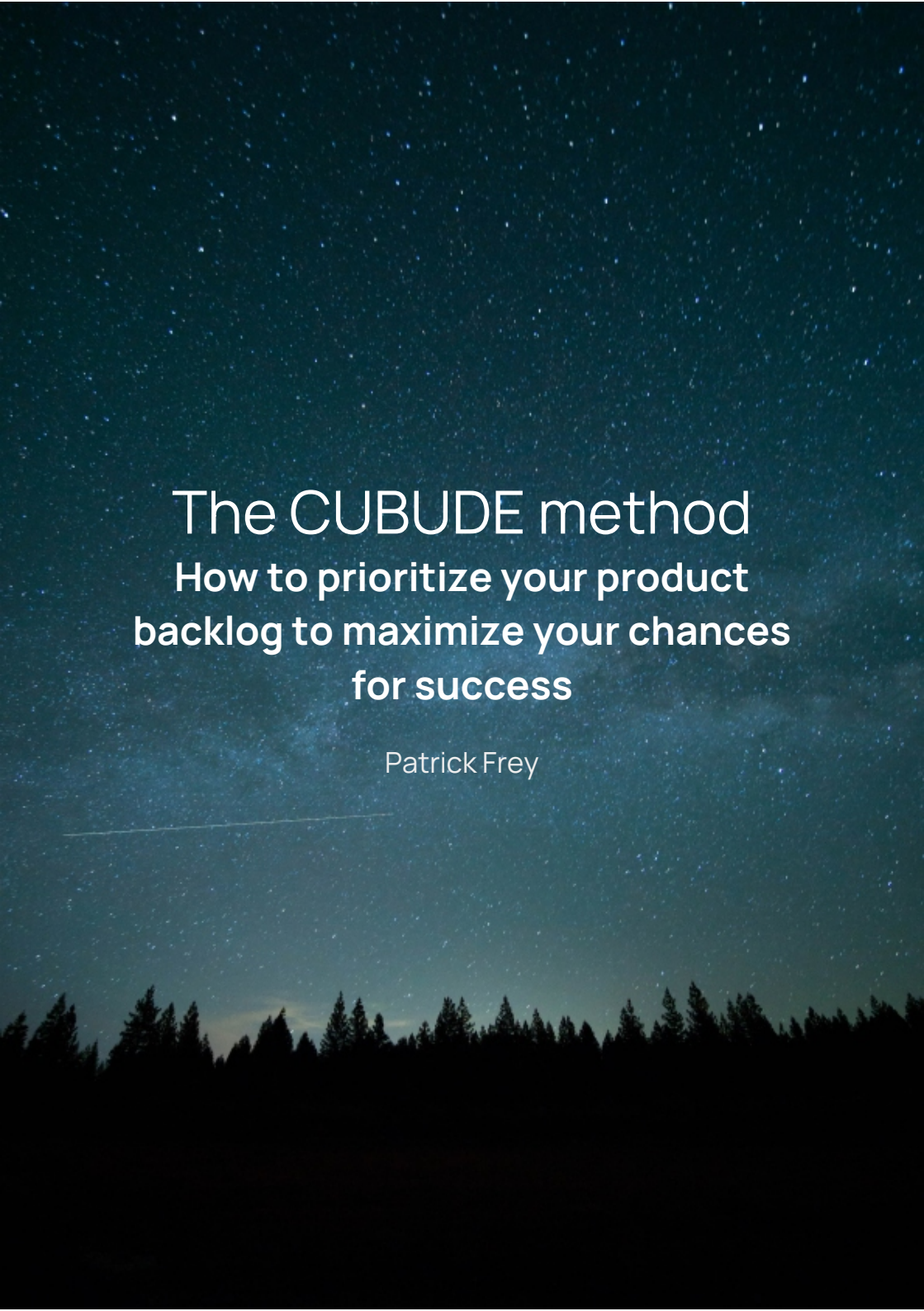

The CUBUDE method

How to prioritize your product backlog to maximize your chances for success

Patrick Frey

Contents

Who should read this book . . . 1

The challenge of prioritizing . . . 3

Where does the approach described in this book come from? . . . 4

The CUBUDE method . . . 6

Example backlog . . . 7

1. Goals and Constraints for Prioritization . . . 9

1.1. Goals . . . 9

1.2. Constraints . . . 10

2. Examining and prioritizing product backlog items from different perspectives . . . 12

2.1. Categorizing backlog items and assigning values . . . 14

2.2. The Customer Perspective . . . 17

2.2.1. The KANO model . . . 17

2.2.2. How can the KANO model be applied to your backlog? . . . 20

2.2.3. How can the KANO model be helpful for priorization? . . . 28

2.3. The Business Perspective . . . 33

2.3.1. Business Value Categories . . . 34

2.3.2. Business Value . . . 36

2.4. The Development Perspective . . . 39

2.4.1. Example for Effort Estimates . . . 42

3. Bringing all Perspectives together . . . 43

3.1. Creating packages of product backlog items for incremental release versions . . . 47

3.1.1. V0: Setting up infrastructure 49
3.1.2. V1..Vn: Initial versions („MVP“) 50
3.1.3. Vn+1 .. Vm: Expanded versions for repeatable business 53
3.1.4. Vm+1: Maintain and expand established product / service 55
3.2. Cross-checking Business Value and Development Efforts 55
3.2.1. Product versions from Business Perspective 57
3.2.2. Product versions from Customer Perspective 57
4. Summary 59
5. Disclaimer 60

Who should read this book

Agile methods such as Scrum and Kanban have been adopted widely over the past years, especially in software development.

In Scrum, for example, the **product backlog** is the central repository of all work that should be done in the future by the team who is working on the product. A backlog is a priorized list of issues such as epics, user stories or tasks.

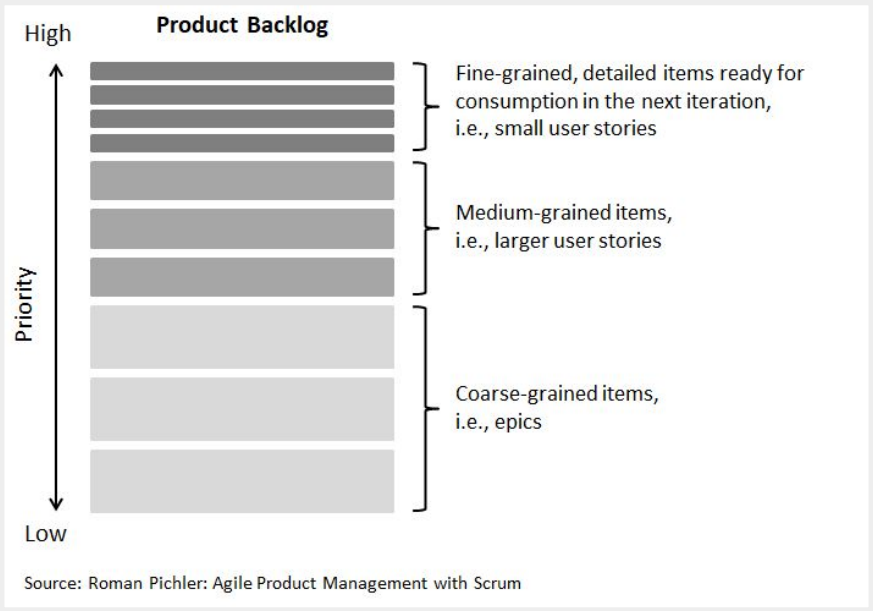

Figure 1. Product Backlog

The product backlog is the information backbone. It enables collaboration between product owner(s) and the development team(s). It is the basis for all kinds of planning activities, such as sprint planning, release planning and roadmap planning. It also provides transparency to stakeholders such as management and

sales.

Maintaining the product backlog is the responsibility of the product owner. **If you are a product owner or have a similar role (e.g., product manager), this book is for you.**

The challenge of prioritizing

In theory, the product backlog is a prioritized list of work items where the first item has the highest priority, followed by second, etc.

But how to prioritize a set of work items? What should actually come first, second, third? Why? What's the motivation and reasoning behind the prioritization? How can you explain or even defend the prioritization? What are the objectives behind the prioritization?

If you find prioritizing your product backlog challenging, this book is for you! It provides a set of methods that have helped several Scrum teams to work efficiently and deliver great value to both internal organization and customers / users.

Where does the approach described in this book come from?

In 2011, I had the opportunity to start and then lead the development activities for a new software-based solution called **EHANDBOOK** [1] at ETAS GmbH (http://www.etas.com/).

During this time, we had spent over 1 year with user research and collected a huge list of requirements (mostly in Excel). When asking the domain experts who formulated the requirements about their priorities, 40 items were priority 1, 30 were priority 2 and 10 were of priority 3.

It was clear that this prioritization was not helpful for us who wanted to build a suitable solution in reasonable time. Rather, it reflected that users had a lot of pain with existing solutions.

The challenge was then to bring the list of requirements into a manageable form such that we could **incrememtally build** a solution.

During this time, we (the EHANDBOOK team) had adopted Scrum as a product development method. (We were the first team at ETAS to officially fully work as an agile team, and we also wanted to succeed in order to convince the internal management.) In my role as product owner, I had the challenge to prioritize the topics and lay out a master plan that would guide us into a successful future.

When searching for suitable methods to analyze and prioritize user wishes, I came across the **KANO model** which I found very helpful. The development team was using planning poker to estimate the user stories.

Since then, our team has conducted roughtly 200 development sprints and over 30 releases. We succeeded to establish a successful product business, and the majority of our users also really likes the product. The approach used to prioritize has thus not failed, but helped us (primarily me) to succeed.

The CUBUDE method

Any good method requires a name such that people can refer to it.

So where does **CUBUDE** come from?

The prioritization method is fundamentally based on examining work items from three different perspectives:

- the **CU**stomer perspective
- the **BU**siness perspective and
- the **DE**velopment efforts perspective

So the name **CUBUDE** is derived by taking the first two letters from each perspective. It is a great aid for memorizing the perspectives more easily and the order in which they should be examined.

Example backlog

To make it easier to follow and understand the steps of the CUBUDE method, we'll use a simple example.

The example is based on a web- or mobile application which provides a user registration and login.

Here's the initial unpriorized backlog of all work items that have been idenfitied:

ID	Backlog Item
1	User registration with email & password
2	Login with email & password
3	Login with Fingerprint
4	Login with FaceID
5	Password reset
6	Usage of Facebook / Twitter user accounts for registration / login
7	User profile: Name
8	User profile: Photo
9	User profile: Video
10	User profile: Appear at #1 position in search results
11	Delete account
12	Admin backend for managing users
13	Setup of new application based on framework XY
14	Setup of development / staging / production environment

[1] EHANDBOOK is a solution for interactive documentation of the application software for automotive embedded control units (ECUs)

Chapter 1. Goals and Constraints for Prioritization

1.1. Goals

Before starting to prioritize, you should be clear about the **goals** you pursue with your project, strategy, etc.

Goals that should be reflected in the prioritization are

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

These goals are the most important goals when it comes to development of new products or services. Without value being delivered to your customers, there will simply be no customer who wants to buy your product. This is a threat for your business as no buyer means no revenue. On the other hand, excessive development efforts for low priority topics will simply cost you a lot of time and money.

The goals are, however, are not automatically aligned. They even can pull in different directions - and this can be a threat to the success of your project or business.

- There can be a **high customer value, but low business value** and/or **high development efforts**. For example, a free lunch is great for a customer, but not substainable for a restaurant business.

- High development efforts can render great results and deliver a lot of customer value, however, you might simply not be able to spent the necessary resources due **to too high cost** or **too long time that it takes to meet your deadline**.

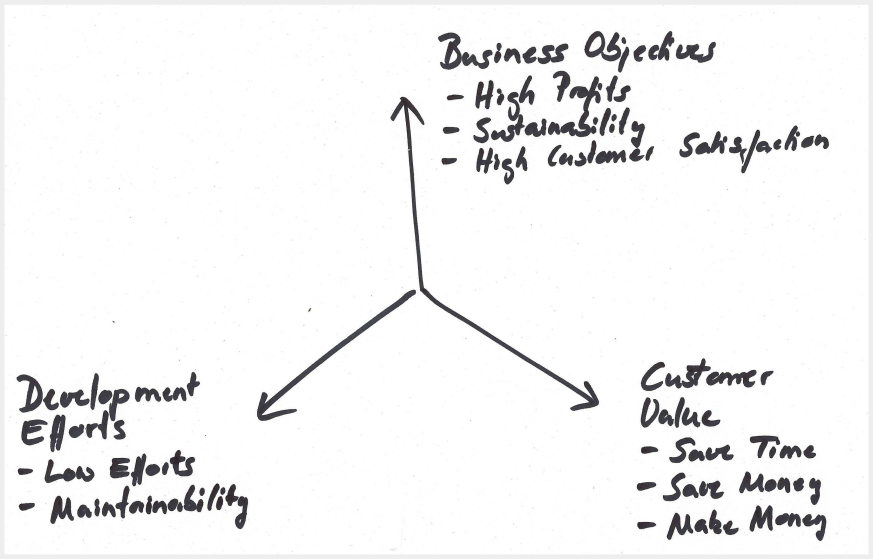

Figure 2. Goals pulling efforts in different directions

In order to have a clear and transparent prioritization, all three goals have to be considered together. Instead of focusing on the goals individually, the **return-on-invest (ROI) in development activities should be maximized**.

1.2. Constraints

The above mentioned goals are often constrained by various aspects such as

- **technical dependencies:** a specific feature can only be built if a certain technical prerequisite is available

- **external dependencies:** the team is dependent on another team or external supplier

These also need to be considered when prioritizing your product backlog. Sometimes you need to take one step back or to the side in order to make two steps ahead.

Chapter 2. Examining and prioritizing product backlog items from different perspectives

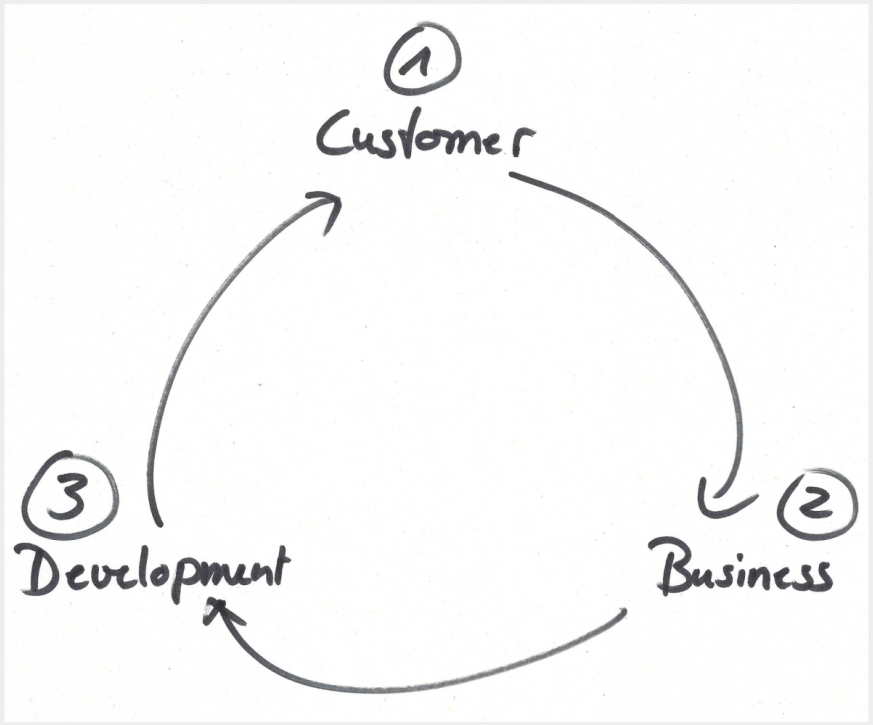

Figure 3. Examining and prioritizing product backlog items from different perspectives

In order to come up with **a good and reasonable priorization**, the items of the product backlog should be inspected from three different perspectives:

1. The **customer perspective**: How your customers would be affected by shipping a specific backlog item.
2. The **business perspective**: How realizing a backlog item presumably affects your business.
3. The **development perspective**: How working on a specific backlog item affects your development.

To enfore customer-centricity, always start with the customer perspective when analyzing product backlog items. Why? Customer-centric organizations are much more successful that other companies who are - for example - technology driven. Take Amazon, for example:

> 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

By taking development perspective last, you'll also not be biased in case of presumably high development efforts and more open to provide some solution the backlog items with high customer and business value.

2.1. Categorizing backlog items and assigning values

Each of the above-mentioned perspectives - the customer perspective, the business perspective and the development perspective - can be assigned to a customer value and business value **category**, or represented by a **value** such as development effort or business value. For customer and business perspective, we'll use categories that help us to analyze and understand customer needs as well as business aspects and impacts.

Why doing so?

The benefit of assigning product backlog issues to categories automatically clusters them such that you have an overview on related items. This creates a taxonomy for the product backlog items. (I like think of the categories as my drawers. I can open a drawer and look into it to find all kinds of items which are akin.).

The benefit of assigning abstract values to items is that it makes them (e.g., in the form of story points or business values) is that it makes them comparable in a relative way.

Both **categorization** and **application of abstract values** together help you to determine direction (=category) and maximize ROI (considering assigned values).

Customer Value is what is typically captured in the form of epics or user stories. To categorize product backlog items according to different customer values, the **Kano model** is used.

Business Value captures what is achieved with the user story.

Who benefits if the user story is delivered? What's the effect on business?

Development Effort is what it takes for the team to work on a product backlog item, deliver its result and consequently deliver value to its target users. It is a mixture of work effort and complexity. In Scrum, the size and complexity of user stories is typically estimated in story points. A method to arrive at an estimation is planning poker.

You can think of the challenge of prioritizing a backlog as solving Rubik's cube, but with different colors. What you want to achieve is to prioritize on items with high business value, high customer value and low development efforts. When these align in one corner of your cube you have found a sweet spot that maximize your chances to succeed.

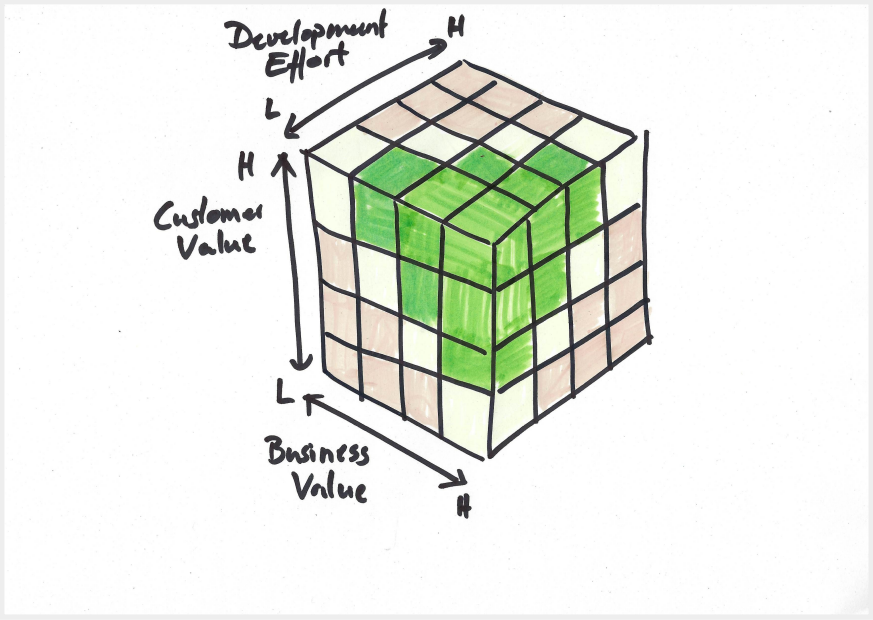

Figure 4. Examining and prioritizing product backlog items from different perspectives

2.2. The Customer Perspective

Focusing on the **customer value** is the most important key success factor for a new solution, product or service.

Customer value is typically being delivered by solving your customers / your users problem(s). The more effective, the better.

A challenge is to find out

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

These questions are difficult to answer, especially from development perspective, but also from product owner / product manager perspective.

2.2.1. The KANO model

The KANO model - also referred to as KANO model of customer satisfaction - provides a very helpful system to classify customer or user wishes according to 5 different categories. The categorization can be used to derive insights into the user wishes and also relate them to the effect of your business.

The KANO model distinguishes 5 levels:

- **Basic characteristics:** These characteristics are so basic and self-evident that users only realize them when they are missing (implicit assumptions). If basic needs are not met, this creates dissatisfaction; if they are met, this doesn't create

satisfaction. The increase in user benefits is rather small in comparison to competitive solutions.

- **Performance characteristics:** Customers and users are aware of them, and they remove dissatisfaction or create satisfaction, depending on the degree of fulfillment.
- **Deligther characteristics:** These are benefit-creating characteristics that the customers or users do not expect per se. They make the product clearly distinguishible from competitor products. A small improvement in this area can lead to disproportionate benefits. The differentiation from competitors might be small, but the benefits can be enormous.
- **Unimportant characteristics:** It is irrelevant for customers or users if such characteristics are present or not. When present, they do not lead to satisfaction, and when not present, they also do not lead to dissatisfaction.
- **Rejection characteristics:** When present, these characteristics lead to dissatisfaction.

	Basic	Performance	Delighter	Unimportant	Rejection
User perspective (Classification from KANO model)	Implicit User is dissatisfied if not fulfilled	User is aware Creates customer value -Removes dissatisfaction	Customer value beyond expectation	User is indifferent / doesn't care	Dissatisfaction if present
Business perspective (Interpretation of KANO model classification)	No customer satisfaction No differen-tiation from competitors	(Some) differentiation from competitors	Can be strong differentiation from competitors	Unimportant for differentation from competitors	To be avoided

2.2.2. How can the KANO model be applied to your backlog?

Start with your existing backlog. I recommend working with spreadsheets such as Microsoft Excel or Google Sheets as it gives you a lot of freedom to work with your items. If you have all your product backlog items in a tool such as Jira, you can create an export to Excel or CSV. In the spreadsheet, simply add a column for "Customer Value".

But where does the actual classification come from? There are in principle three approaches:

1. **User research:** To obtain the necessary insights for the classification, you can (and should!) conduct user interviews and ask your target customers / users.
2. **Applying domain expertise:** As a domain expert, you can apply your existing knowledge and apply it to the backlog. For items which are not completely clear, you can leave them out or assign a value and mark them.
3. **Best educated guess:** Yes, you can guess. It's not strongly recommended, but can help you to fill some gaps. Better than guessing, however, is consulting a domain expert or - at best - conducting user research.

Go through each of your items and assign a customer value category: **Basic**, **Performance**, **Deligther**, **Unimportant**, **Rejection**. Sometimes, it might not be clear if an item belongs to one or multiple categories. In your spreadsheet, simply list all categories that you think are applicable. When sorting / filtering according to Customer Value, the marked items will show up as both.

ID	Backlog Item	Customer Value
1	User registration with email & password	Basic
2	Login with email & password	Basic
3	Login with Fingerprint	Performance
4	Login with FaceID	Performance / Deligther
5	Password reset	Basic / Rejection Point
6	Usage of Facebook / Twitter user accounts for registration / login	Performance / Deligther
7	User profile: Name	Basic
8	User profile: Photo	Performance
9	User profile: Video	Deligther / Unimportant
10	User profile: Appear at #1 position in search results	Deligther
11	Delete account	Basic / Rejection Point
12	Admin backend for managing users	Unimportant
13	Setup of new application based on framework XY	Unimportant
14	Setup of development / staging / production environment	Unimportant

Conducting user research / user interviews to support the KANO model classification

The 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 ...?"
2. "What would you say if the product / solution could not do /

would not have …?"

The first one is a positive formulation in order to identify functional needs while the second one is a negative formulation to identify dysfunctional needs.

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

	Functional	Disfunctional
	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		

Depending on the answers of your interviewed customers / users, you can derive a classification.

Basic needs

Basic needs of users can be identified when they are **very happy** or **take it for granted** if a product/solution provides a certain capability, however, they would be **very annoyed if it is missing**.

> A basic capability of an umbrella is that it protects you from getting wet in the rain. This is something which is typically taken for granted, but very annoying if it is not the case.

Table 1. Customer Value = Basic

	Functional	Disfunctional
	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

Performance

Performance attributes can be identified when users are **very happy** if a product/solution provides a certain capability, however, they would **barely accept if not**.

> A performance attribute of an umbrella would be that its cloth is water-resistent and dries very fast. This is something where users are typically very happy about, but would barely accept if a used umbrella takes hours to dry from soaked water.

Table 2. Customer Value = Performance

	Functional	Disfunctional
	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		

Deligther

Deligther attributes can be identified when users are **very happy** if a product/solution provides a certain capability, however, they **don't really care if it is not present**.

> A deligther attribute of an umbrella would be that unfolds automatically on the press of a button. This is something where users are typically very happy about, but opening an umbrella works also manually.

Table 3. Customer Value = Deligther

	Functional	Disfunctional
	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		

Unimportant

An unimportant attribute is when users **don't care** if a product/solution provides a certain capability or not.

> A unimportant attribute - at least for many users - of an umbrella would whether its shaft is made from wood or metal.

Table 4. Customer Value = Unimportant

	Functional	Disfunctional
	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		

Rejection

A rejection attribute is the opposite of a deligther, i.e. when would be **very annoyed** if a product/solution provides a certain capability, and they would be very happy or take it for granted if the capability is not present.

> A attribute where most users would reject an umbrella is when it cannot be collapsed, but always stays open.

Table 5. Customer Value = Rejection

	Functional	Disfunctional
	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	

2.2.3. How can the KANO model be helpful for priorization?

The insights that you have gained can now be used for priorization. Ultimately, the priorization also depends on the goals that you want to achieve:

- **Validate MVP / new concept:** If your goal is to validate an MVP or a concept for a new feature, your focus should be on items are marked with Performance or Deligther - especially if there is still a lot of uncertainty. Rationale: You want to confirm that your idea provides value to your tentative customers / users.
- **Improve Quality / UX:** If your goal is to improve the perceived quality of your product or service or to achieve an improvement of the user experience, you should focus on items marked with Rejection. Rationale: By getting rid of rejection points that your existing users complain about, you can improve your overall net promotor score (NPS) by reducing the number of distractors. This is sometimes much easier than increasing the number of attractors, especially when rejection points are generic and also apply to your existing attractors.
- **Derive a roadmap of meaningful product increments**: Creating a new product or service typically is done in multiple steps and with incremental releases. You also want to ensure that new customers and users can grasp easily that your product or service provides value. Hence, for any product area or epic, you should identify the necessary Basic and Deligther items. If already required, you can include Performance items. Note that each release should be consistent in itself, i.e. you do not want to deliver any half-baked features (e.g., a login without a logout).

Example for validation of MVP / new concepts:

When filtering the example backlog for items categorized with Deligther, the following list shows up:

ID	Backlog Item	Customer Value
6	Usage of Facebook / Twitter user accounts for registration / login	Performance / Deligther
10	User profile: Appear at #1 position in search results	Deligther
9	User profile: Video	Deligther / Unimportant
4	Login with FaceID	Performance / Deligther

Login with Facebook / Twitter or FaceID is already very common in other web or mobile applications, so there is little uncertainty that these additional capabilities provide additional value. Using videos in user profiles is, however, an uncommon feature which is not present in any widely used web or mobile apps. This, however, doesn't mean that it is not valuable for your customers or users. The same applies for users being able to appear at position #1 in searches within your web or mobile app.

Example for Quality / UX improvement

Assuming you already have a product with basic user registration and login, to improve the user experience for existing users, the self-service possibilities to reset the password or delete the account are of highest priority.

ID	Backlog Item	Customer Value
5	Password reset	Basic / Rejection Point
11	Delete account	Basic / Rejection Point

Example for roadmap priorization

The following table shows a tentative backlog priorization where the items are assigned to different product increments. In V0, the dev team creates the necessary infrastructure to develop, test and ship the new web or mobile application. V1 then contains basic features for new users to register and login with an email and password of their choice. For personalization, V1 also already provides the possibility to provide a username. V2 extends V1 with a password reset possibility as this would frustrate users. Adding provile pictures makes the product more attractive to the user whilst having an admin backend is relevant for the ops team. The next increment - V3 - provides the possibility to utilize existing Facebook or Twitter accounts with the product. Also, users can get unregister themselves by deleting their account - something that customer service had to take care of earlier. A deligther for is to be listed #1 position in the user search. Note that this could be something that some users would even pay for. In V4 and V5, the product is further enhanced by providing login with fingerprints or FaceID, or using videos as user profiles instead of a photo.

ID	Backlog Item	Customer Value	Release
13	Setup of new application based on framework XY	Unimportant	V0
14	Setup of development / staging / production environment	Unimportant	V0
1	User registration with email & password	Basic	V1
2	Login with email & password	Basic	V1
7	User profile: Name	Basic	V1
5	Password reset	Basic / Rejection Point	V2

ID	Backlog Item	Customer Value	Release
8	User profile: Photo	Performance	V2
12	Admin backend for managing users	Unimportant	V2
6	Usage of Facebook / Twitter user accounts for registration / login	Performance / Deligther	V3
10	User profile: Appear at #1 position in search results	Deligther	V3
11	Delete account	Basic / Rejection Point	V3
3	Login with Fingerprint	Performance	V4
9	User profile: Video	Deligther / Unimportant	V4
4	Login with FaceID	Performance / Deligther	V5

Further resources:

- Wikipedia: https://en.wikipedia.org/wiki/Kano_model (English) and http://de.wikipedia.org/wiki/Kano-Modell (German)

2.3. The Business Perspective

Another very important aspect for the priorization of your product backlog items is the business perspective.

Businesses require revenue to fund their activities such as development and operations. These are obtained from existing and new customers who purchase or subscribe to your product, solution or service.

The key success factor for **new products or services** is that the need to provide enough value to paying customers such that the business can be operated - and thus works.

Note that the customer value (or customer benefit) is related to the business value, but not the same as they have different perspectives:

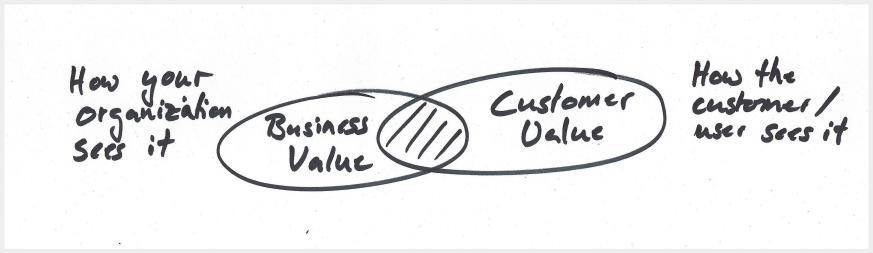

Figure 5. Customer value and business value

Key questions from business perspective are:

- How can business value be measured?
- How can business value be maximized?
- When is the right time for a feature?
- How to achieve fast time-to-market?

2.3.1. Business Value Categories

First of all, it makes sense to categorize the items in your product backlog according to the effect that it would have on your business when the item gets realized / implemented.

In principle, there are four 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, up-grade 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, ...)

- The distintion between new business and retainment often goes hand in hand with Basic and Performance needs of customers / users
- Items which are classified as Unimportant from customer / user perspective often have a benefit to your Operationel Efficiency. This can be short term or long term.

Example for Business Value Categorization

ID	Backlog Item	Business Value Category
1	User registration with email & password	1 - New Business
2	Login with email & password	1 - New Business
3	Login with Fingerprint	2 - Retainment
4	Login with FaceID	2 - Retainment
5	Password reset	4 - Operational Efficiency
6	Usage of Facebook / Twitter user accounts for registration / login	1 - New Business
7	User profile: Name	2 - Retainment
8	User profile: Photo	2 - Retainment
9	User profile: Video	2 - Retainment
10	User profile: Appear at #1 position in search results	3 - Up sell
11	Delete account	4 - Operational Efficiency
12	Admin backend for managing users	4 - Operational Efficiency
13	Setup of new application based on framework XY	4 - Operational Efficiency
14	Setup of development / staging / production environment	4 - Operational Efficiency

2.3.2. Business Value

As a next step, you can assign a business value number to the backlog items.

The approach for assigning business value numbers is inspired by Andrea Tomasini (2007).

It is an abstract way to measure business value.

We suggest to utilize the following numbers: 100, 200, 300, 500, 800, 1300, 2000, 3000. You could also extend the numbers further.

Similar as Story Points which are used for estimating development effort, the sequence of numbers is inspired by the Fibonacci sequence, multiplied by 100. The multiplication factor of 100 is useful to later relate the business value numbers to effort estimates in story points. Through this, a return-on-invest number can be calculated which is useful for further priorization. (Note that 2000 and 3000 are not directly related to Fibonacci numbers, but rounded.)

How to apply Business Value numbers

The purpose of using abstract numbers is to be able to make relative comparisons between the items. And: the higher the number, the higher the (estimated) business value.

We recommend to go through your items one by one per Business Value category. Through this, you can focus on a specific business aspect (e.g., New Business). This will make it easier to relatively compare the items and assign a meaningful business value number. If you are using a spreadsheet in Google Sheets or Microsoft Excel, you can filter your backlog to only show the items

from a specific Business Value Category.

Example for Business Value assignment

The following except shows the backlog items filtered according to the Business Value category "New Business". In the example, we have assigned a business value of 3000 to the user registration with email and password. This value is even higher than the login with email and password itself as the email addresses of customers / users are valuable data, e.g. for product marketing via email. Compared to the user registration and login with email items, the value of user registration with Facebook / Twitter is less.

ID	Backlog Item	Business Value Category	Business Value
1	User registration with email & password	1 - New Business	3000
2	Login with email & password	1 - New Business	2000
6	Usage of Facebook / Twitter user accounts for registration / login	1 - New Business	1200

Our complete example now looks like this:

ID	Backlog Item	Business Value Category	Business Value
1	User registration with email & password	1 - New Business	3000
2	Login with email & password	1 - New Business	2000
3	Login with Fingerprint	2 - Retainment	800
4	Login with FaceID	2 - Retainment	1200
5	Password reset	4 - Operational Efficiency	1200

ID	Backlog Item	Business Value Category	Business Value
6	Usage of Facebook / Twitter user accounts for registration / login	1 - New Business	1200
7	User profile: Name	2 - Retainment	2000
8	User profile: Photo	2 - Retainment	1200
9	User profile: Video	2 - Retainment	1200
10	User profile: Appear at #1 position in search results	3 - Up sell	2000
11	Delete account	4 - Operational Efficiency	800
12	Admin backend for managing users	4 - Operational Efficiency	1200
13	Setup of new application based on framework XY	4 - Operational Efficiency	2000
14	Setup of development / staging / production environment	4 - Operational Efficiency	1200

2.4. The Development Perspective

In Scrum, knowing the effort it takes to deliver some of backlog items is crucial for sprint and release planning activities.

For estimating the effort of backlog items, story points (SP) are widely used. They are also used in our approach for backlog priorization.

Story Points can be applied to Themes, Epics as well as User Stories.

They are an abstract measure that capture both effort and complexity (i.e., uncertainty) of an item.

- They are abstract numbers without a unit (no direct relation to hours, days etc.)
- They allow for relative comparison of product backlog items concerning the effort

Some common pitfalls:

- 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.

Story Points are captured through numbers which are inspired by the Fibonacci sequence: 0, 1, 2, 3, 5, 8, 13, 20, 40, 70, 100, 200, 500, 1000, …

- Note that above 20, the numbers do not correspond to the Fibinacci sequence any more as this gives some false impression on how exact the team can estimate the effort.

- There are even Planning Poker cards

The effort is typically estimated by the development team itself, i.e. a group of people.

1. Product Owner (you) explains the Theme, Epic, User Story
2. A (hopefully) vital discussion arises where team members ask questions that the PO and other team members can (hopefully) clarify. Typically, senior developers and architects can give some guidance to the team how the user story can be realized. The goal is to drive out uncertainty and identify pending risks.
3. The whole team estimates, each team member for himself/herself. The estimates are then revealed. The goal is to achieve a consensus on the effort estimate.
4. If numbers divert, the team members with highest and lowest numbers explain their rationale. The team discusses and agrees on a effort estimate.

Some hints for team hygiene: Estimating work effort in an open space requires a trustful relationship among all team members (including the PO).

- As PO, I recommend not to 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 the user story to flesh out certain and uncertain parts.
- Work together, not against each other.

Over the time, the team will get a feeling for the story points that are applicable to User Stories, Epics and Themes.

For example, everything between 0 and 13 can be considered an effort estimate for a User Story while estimates in the range of 20-100 relate to larger Epics. Number beyond 200 can be considered as themes.

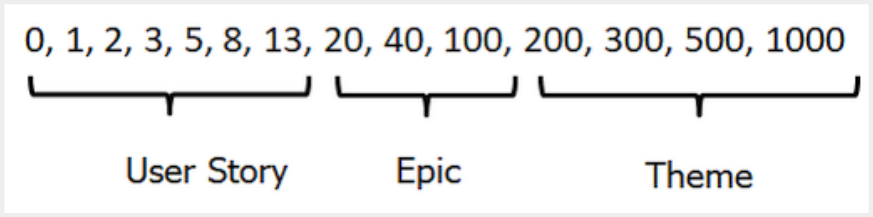

Figure 6. Fibinacci sequence for estimating user stories

- User Stories should be completely clear in scope, not have any uncertainty / known risks left and basically just be "effort". This is ideal as such user stories can then be delivered in one sprint by the team, leading to a viable product increment.
- Epics typically require more effort and there can be quite some uncertainty left. They need to be further analyzed such that they can be broken down into individual user stories. Nevertheless, epics should fit into a release (depending on your release cadence).
- Themes are often so large that more substantial work is required to deliver them completely.

2.4.1. Example for Effort Estimates

The following table shows our example backlog, now with development effort for each item in Story Points.

ID	Backlog Item	Development Effort (Story Points)
1	User registration with email & password	20
2	Login with email & password	13
3	Login with Fingerprint	40
4	Login with FaceID	20
5	Password reset	8
6	Usage of Facebook / Twitter user accounts for registration / login	8
7	User profile: Name	5
8	User profile: Photo	8
9	User profile: Video	13
10	User profile: Appear at #1 position in search results	8
11	Delete account	5
12	Admin backend for managing users	20
13	Setup of new application based on framework XY	13
14	Setup of development / staging / production environment	20

The numbers allow for relative comparison. For example, adding a name to a user profile (5 SP) is less effort than adding a photo (8 SP) or video (13 SP). And adding a photo (8 SP) is less effort than adding a video (13 SP); while the team has experience in how to handle photos, there is more uncertainty for how to handle videos.

Chapter 3. Bringing all Perspectives together

We can now bring all perspectives together and derive a priorization that best supports our goals.

The example backlog now has information and data that captures the customer, business and development perspective as follows:

ID	Backlog Item	Customer Value	Business Value Category	Business Value	Development Effort
1	User registration with email & password	Basic	1 - New Business	3000	20
2	Login with email & password	Basic	1 - New Business	2000	13
3	Login with Fingerprint	Performance	2 - Retainment	800	40
4	Login with FaceID	Performance / Deligther	2 - Retainment	1200	20
5	Password reset	Basic / Rejection Point	4 - Operational Efficiency	1200	8
6	Usage of Facebook / Twitter user accounts for registration / login	Performance / Deligther	1 - New Business	1200	8
7	User profile: Name	Basic	2 - Retainment	2000	5
8	User profile: Photo	Performance	2 - Retainment	1200	8

ID	Backlog Item	Customer Value	Business Value Category	Business Value	Development Effort
9	User profile: Video	Deligther / Unimportant	2 - Retainment	1200	13
10	User profile: Appear at #1 position in search results	Deligther	3 - Up sell	2000	8
11	Delete account	Basic / Rejection Point	4 - Operational Efficiency	800	5
12	Admin backend for managing users	Unimportant	4 - Operational Efficiency	1200	20
13	Setup of new application based on framework XY	Unimportant	4 - Operational Efficiency	2000	13
14	Setup of development / staging / production environment	Unimportant	4 - Operational Efficiency	1200	20

As both business value and development effort are estimated in abstract numbers, it allows us to consider them in relation to each other and calculate a "Return on Investment" figure. What the ROI basically expresses is how much business value is created when investing 1 story point of effort.

ROI = Business Value / Development Effort

The higher the ROI, the more efficiently your scarce development resource - i.e. available development time - is utilized.

ID	Backlog Item	Customer Value	Business Value Category	Business Value	Development Effort	Return on Invest (ROI) Ratio
1	User registration with email & password	Basic	1 - New Business	3000	20	150
2	Login with email & password	Basic	1 - New Business	2000	13	154
3	Login with Fingerprint	Performance	2 - Retainment	800	40	20
4	Login with FaceID	Performance / Deligther	2 - Retainment	1200	20	60
5	Password reset	Basic / Rejection Point	4 - Operational Efficiency	1200	8	150
6	Usage of Facebook / Twitter user accounts for registration / login	Performance / Deligther	1 - New Business	1200	8	150
7	User profile: Name	Basic	2 - Retainment	2000	5	400
8	User profile: Photo	Performance	2 - Retainment	1200	8	150
9	User profile: Video	Deligther / Unimportant	2 - Retainment	1200	13	93
10	User profile: Appear at #1 position in search results	Deligther	3 - Up sell	2000	8	250
11	Delete account	Basic / Rejection Point	4 - Operational Efficiency	800	5	160

ID	Backlog Item	Customer Value	Business Value Category	Business Value	Development Effort	Return on Invest (ROI) Ratio
12	Admin backend for managing users	Unimportant	4 - Operational Efficiency	1200	20	60
13	Setup of new application based on framework XY	Unimportant	4 - Operational Efficiency	2000	13	154
14	Setup of development / staging / production environment	Unimportant	4 - Operational Efficiency	1200	20	60

The actual prioritization requires now working with the backlog.

3.1. Creating packages of product backlog items for incremental release versions

With CUBUDE, you to create consistent packages of backlog items that are shipped in incremental product versions: V0, V1, V2, V3, …

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

Assumptions:

- For each product increment, the development team is capable of delivering the same amount of effort.
- In our example, we assume that the overall effort that the dev team can deliver for a release is 35-40 SP.

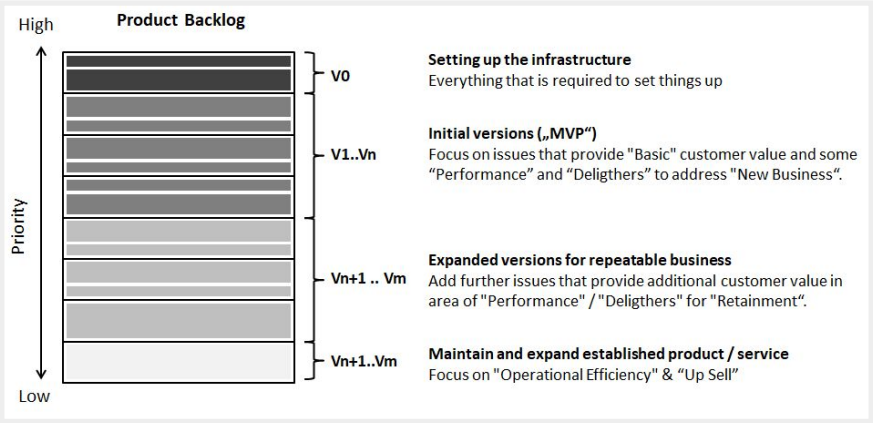

Figure 7. Product Backlog prioritized with CUBUDE method

3.1.1. V0: Setting up infrastructure

Steps:

1. Filter Business Value Category for "Operational Efficiency".
2. Identify items that are required for setting up the necessary infrastructure that is required for the development activities and assign them to version V0

ID	Backlog Item	Customer Value	Business Value Category	Business Value	Development Effort	Return on Invest (ROI) Ratio	Version
5	Password reset	Basic / Rejection Point	4 - Operational Efficiency	1200	8	150	
11	Delete account	Basic / Rejection Point	4 - Operational Efficiency	800	5	160	
12	Admin backend for managing users	Unimportant	4 - Operational Efficiency	1200	20	60	
13	**Setup of new application based on framework XY**	Unimportant	**4 - Operational Efficiency**	2000	13	154	**V0**
14	**Setup of development / staging / production environment**	Unimportant	**4 - Operational Efficiency**	1200	20	60	**V0**

3.1.2. V1..Vn: Initial versions („MVP“)

Steps:

1. Filter Business Value Category for "New Business"
2. Identify items with "Basic" customer value and assign them to version V1

Cross-check overall effort assigned to V1. If aggregated efforts exceed the capacity for one version, start a new version number (V2, V3, etc.)

ID	Backlog Item	Customer Value	Business Value Category	Business Value	Development Effort	Return on Invest (ROI) Ratio	Version
1	User registration with email & password	**Basic**	1 - New Business	3000	20	150	**V1**
2	Login with email & password	**Basic**	1 - New Business	2000	13	154	**V1**
6	Usage of Facebook / Twitter user accounts for registration / login	Performance / Deligther	1 - New Business	1200	8	150	

1. Add filter of Business Value Category for "Retainment", "Up Sell" and "Operational Efficiency"
2. Sort the backlog items according to the calculated ROI value
3. Pick items with high ROI that fit into a release package

ID	Backlog Item	Customer Value	Business Value Category	Business Value	Development Effort	Return on Invest (ROI) Ratio	Version
7	User profile: Name	Basic	**2 - Retainment**	2000	5	400	**V1**
10	User profile: Appear at #1 position in search results	Deligther	**3 - Up sell**	2000	8	250	**V2**
11	Delete account	Basic / Rejection Point	**4 - Operational Efficiency**	800	5	160	**V2**
2	Login with email & password	Basic	1 - New Business	2000	13	154	V1
13	Setup of new application based on framework XY	Unimportant	4 - Operational Efficiency	2000	13	154	V0
1	User registration with email & password	Basic	1 - New Business	3000	20	150	V1
5	Password reset	Basic / Rejection Point	**4 - Operational Efficiency**	1200	8	150	**V2**

ID	Backlog Item	Customer Value	Business Value Category	Business Value	Development Effort	Return on Invest (ROI) Ratio	Version
6	Usage of Facebook / Twitter user accounts for registration / login	Performance / Deligther	1 - New Business	1200	8	150	
8	User profile: Photo	Performance	2 - Retainment	1200	8	150	**V2**
9	User profile: Video	Deligther / Unimportant	2 - Retainment	1200	13	93	
4	Login with FaceID	Performance / Deligther	2 - Retainment	1200	20	60	
12	Admin backend for managing users	Unimportant	4 - Operational Efficiency	1200	20	60	
14	Setup of development / staging / production environment	Unimportant	4 - Operational Efficiency	1200	20	60	V0
3	Login with Fingerprint	Performance	2 - Retainment	800	40	20	

3.1.3. Vn+1 .. Vm: Expanded versions for repeatable business

Next iterations: Identify items for V3, V4.

ID	Backlog Item	Customer Value	Business Value Category	Business Value	Development Effort	Return on Invest (ROI) Ratio	Version
7	User profile: Name	Basic	2 - Retainment	2000	5	400	V1
10	User profile: Appear at #1 position in search results	Deligther	3 - Up sell	2000	8	250	V2
11	Delete account	Basic / Rejection Point	4 - Operational Efficiency	800	5	160	V2
2	Login with email & password	Basic	1 - New Business	2000	13	154	V1
13	Setup of new application based on framework XY	Unimportant	4 - Operational Efficiency	2000	13	154	V0
1	User registration with email & password	Basic	1 - New Business	3000	20	150	V1
5	Password reset	Basic / Rejection Point	4 - Operational Efficiency	1200	8	150	V2

ID	Backlog Item	Customer Value	Business Value Category	Business Value	Development Effort	Return on Invest (ROI) Ratio	Version
6	Usage of Facebook / Twitter user accounts for registration / login	Performance / Deligther	1 - New Business	1200	8	150	**V3**
8	User profile: Photo	Performance	2 - Retainment	1200	8	150	V2
9	User profile: Video	Deligther / Unimportant	2 - Retainment	1200	13	93	**V4**
4	Login with FaceID	Performance / Deligther	2 - Retainment	1200	20	60	**V4**
12	Admin backend for managing users	Unimportant	4 - Operational Efficiency	1200	20	60	**V3**
14	Setup of development / staging / production environment	Unimportant	4 - Operational Efficiency	1200	20	60	V0

3.1.4. Vm+1: Maintain and expand established product / service

V5

ID	Backlog Item	Customer Value	Business Value Category	Business Value	Development Effort	Return on Invest (ROI) Ratio	Version
3	Login with Fingerprint	Performance	2 - Retainment	800	40	20	**V5**

3.2. Cross-checking Business Value and Development Efforts

Having business values and development efforts expressed in numbers and having all information available in a spreadsheet (Google Sheets, Microsoft Excel) allows to coduct several cross-checks:

- Does the development effort for a version exceed the maximum capacity? In your spreadsheet, you can simply sum up all story points of all items assigned to a specific version.
- Is the prioritization in such a way that business value is created as early as possible?

Business Value	Development Effort	Return on Invest (ROI) Ratio	Version
3200	33	214	V0
7000	38	704	V1

Business Value	Development Effort	Return on Invest (ROI) Ratio	Version
5200	29	710	V2
2400	28	210	V3
2400	33	153	V4
800	40	20	V5

3.2.1. Product versions from Business Perspective

You can derive insights from Business Perspective such as where the focus of a specific release is set.

	Version					
Customer Value	V0	V1	V2	V3	V4	V5
1 - New Business	0	2	0	1	0	0
2 - Retainment	0	1	1	0	2	1
3 - Up Sell	0	0	1	0	0	0
4 - Operational Efficiency	2	0	2	1	0	0
Sum	2	3	4	2	2	1

Figure 8. Product versions from Business Perspective

3.2.2. Product versions from Customer Perspective

Additionally, you can derive insights on which customer values the individual versions focus.

In our example, in V0, only "Unimportant" items are delivered. Hence it does not make sense to release this version to any user yet.

As V1 focuses on some "Basic" items, this version might also not be sufficient to be provided to any users. With V2, however, some "Performance" and "Deligther" aspects are provided, and "Rejection Points are removed". This qualifies for a good first MVP that could be shipped to initial users. V3, V4 and V5 add further "Performance" and "Deligthers".

	Version					
Customer Value	**V0**	**V1**	**V2**	**V3**	**V4**	**V5**
Basic	0	3	2	0	0	0
Performance	0	0	1	1	1	1
Deligther	0	0	1	1	2	0
Unimportant	2	0	0	1	1	0
Rejection Point	0	0	2	0	0	0
Sum	2	3	6	3	4	1

Figure 9. Product versions from Customer Perspective

Chapter 4. Summary

Prioritizing work items of a product backlog **requires substantial efforts** and is **not an easy task**.

The **CUBUDE 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**.
- 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**.

Chapter 5. Disclaimer

- **There exist numerous other methods** that can be used for prioritizing a backlog.
- The CUBUDE method is **not a silver bullet** and does not save you from failure.
- Following a systematic approach helps you to **maximize your odds of success**.