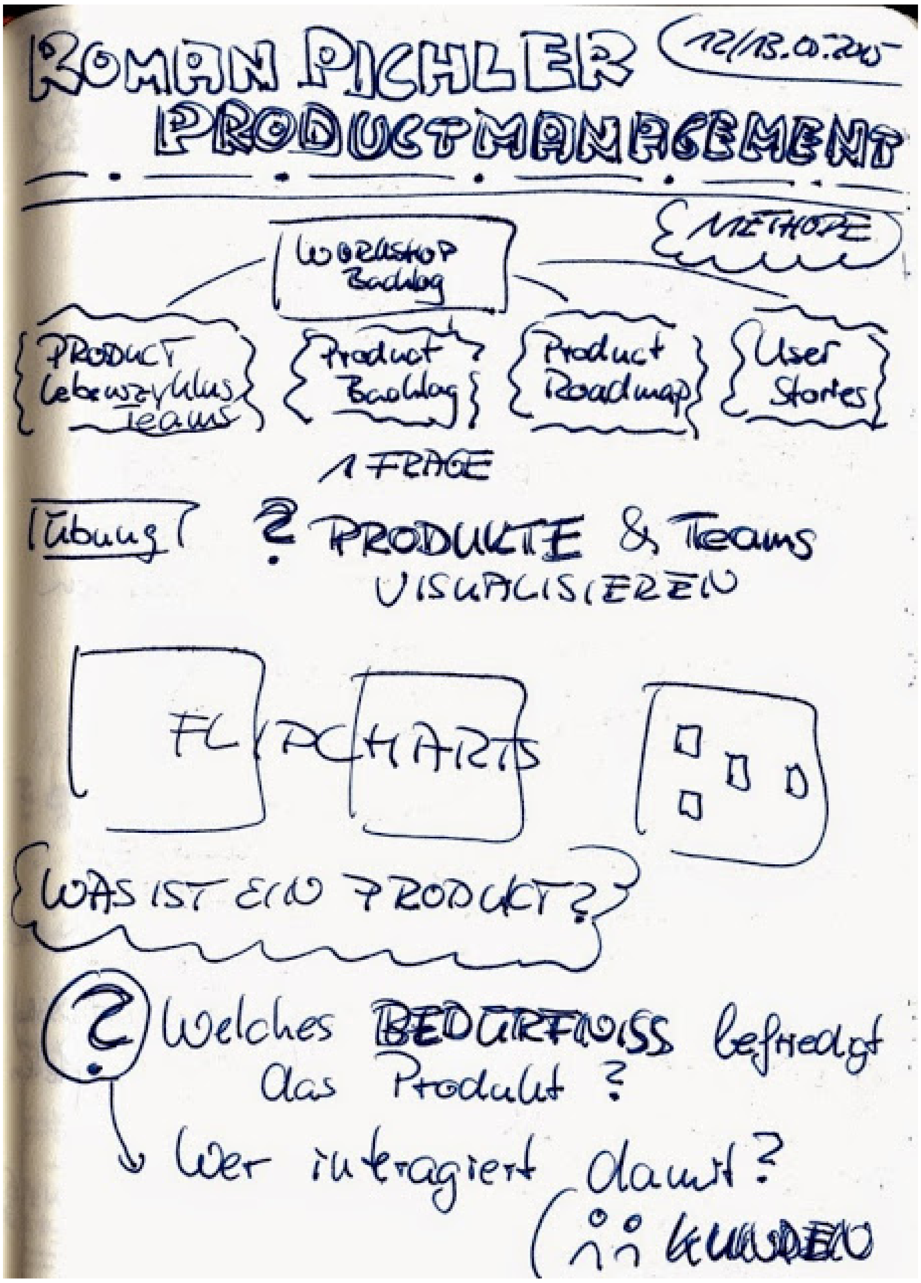
Awesome product management with Roman Pichler – Roadmaps, Products, Personas, Teams – your next training?

ontheagilepath.net/2015/05/awesome-product-management-with-roman-pichler-roadmaps-products-personas-teams-your-next-training.html

By Sebastian Radics

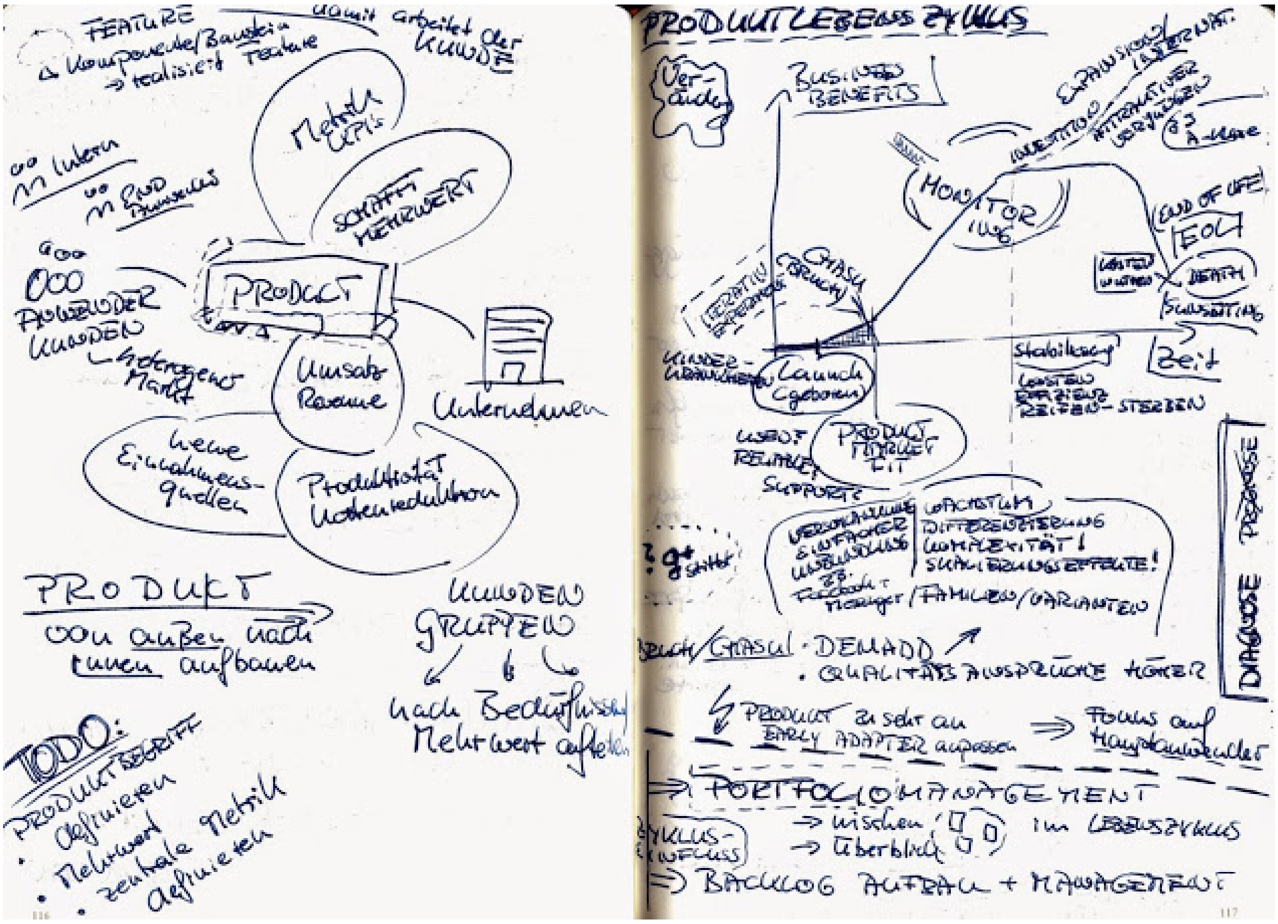
This week I had the opportunity to join an awesome two day workshop – PRODUCTMANAGEMENT – facilitated by Roman Pichler.
High intensity input regarding Products, Roadmaps, Stories, Teams, Personas and Product Management. WOW. In this post I share my Sketchnotes (mainly german but maybe some pictures help all EN readers too).
I try to highlight some key insights so that you can check if you should join this path of experience too.



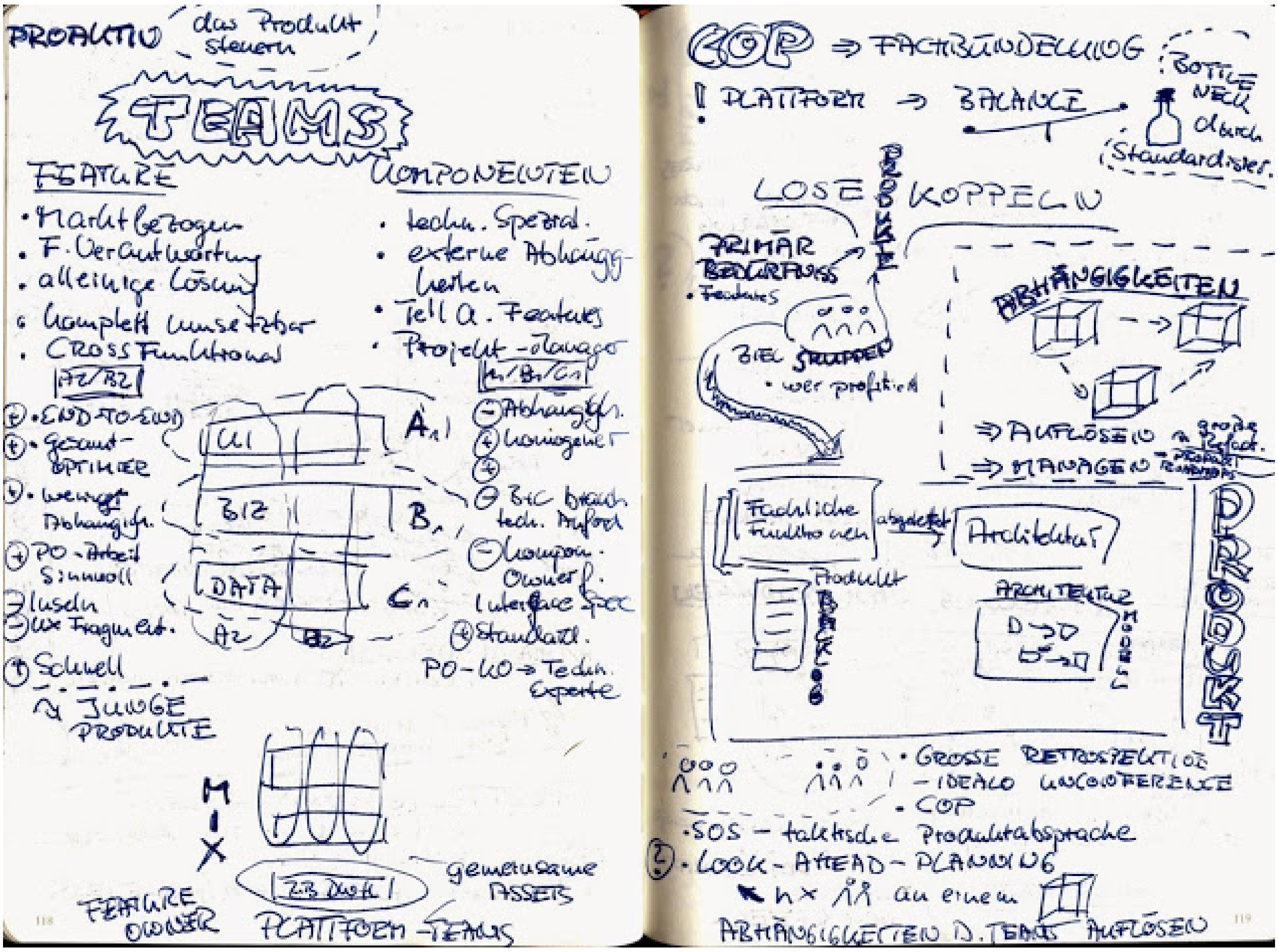
For a product – ask:

- What **needs** does the product address?
- Who interacts with the product?

By using a visualization what products you are building and how your teams are structured you can explore whether all share the same understanding of what a product is. It will for sure challenge your status quo.



Build your product from outside to inside. Structure it by customer groups and check for common needs to address.
A product consists of features (parts the user interacts with) and components implementing the features. By a product you generate revenue, increase productivity/reduce costs, it creates additional value and is used by a heterogenous user market.
We need to consider the product life cycle – with the phases -launch, product market fit, growth, end of life or huge change and adaption to initialize a new growth cycle.
Depending on where the product is positioned in the life cycle curve you need to apply different management and process strategies.
Starting with highly experimental environments moving to a more standardized and productivity+efficiency focused environment.
Part of portfolio management is to balance all products and life cycles (not having to much products in EOL)

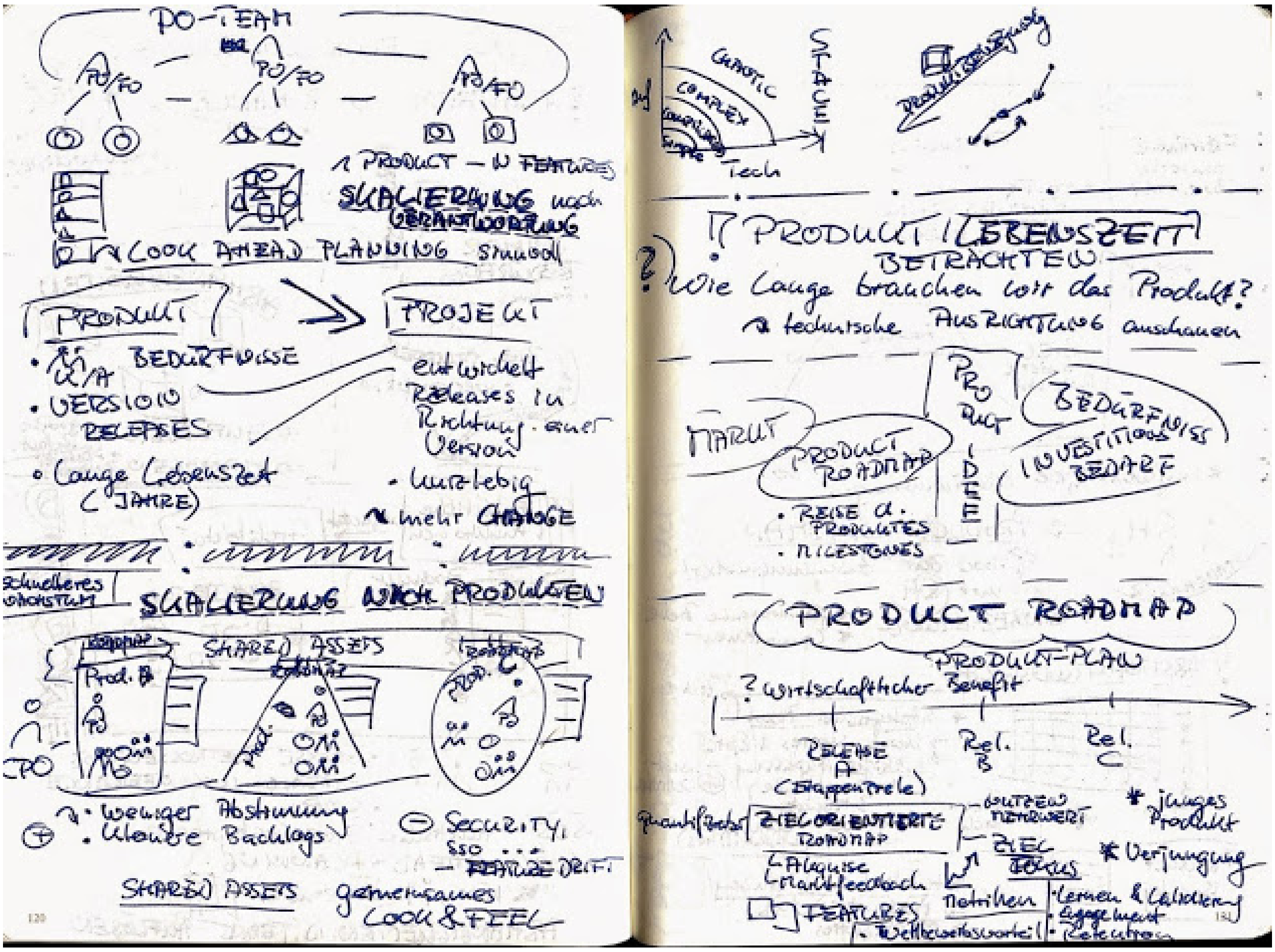


Consider the difference between feature teams and component teams and pros/cons for both team variants. Applying a team mix for common assets (and e.g. building platform teams) can be an option too.

Aim for loosely coupling of products.

Resolve dependencies or apply proper management of dependencies between products. Use e.g. company wide retrospectives (with delegates from every product) and look ahead planning. On tactical level Scrum of Scrums are useful.

Based on goals and needs for the product – you generate the product backlog with functional aspects and derive an architecture model guiding the technical implementation.



A project lifetime is much smaller than a product lifetime. Projects are used to implement a release of a product.

An option is to scale by responsibilities. Have a product and n feature teams building that product. A chief product owner takes the responsibility for the product. Product owners take the responsibility for features. All work with a common backlog.

For faster growth: **Scale by products.** Split your product in multiple ones. Every products has an own product backlog and product owner. The chief product owner keeps the overview on the product line.

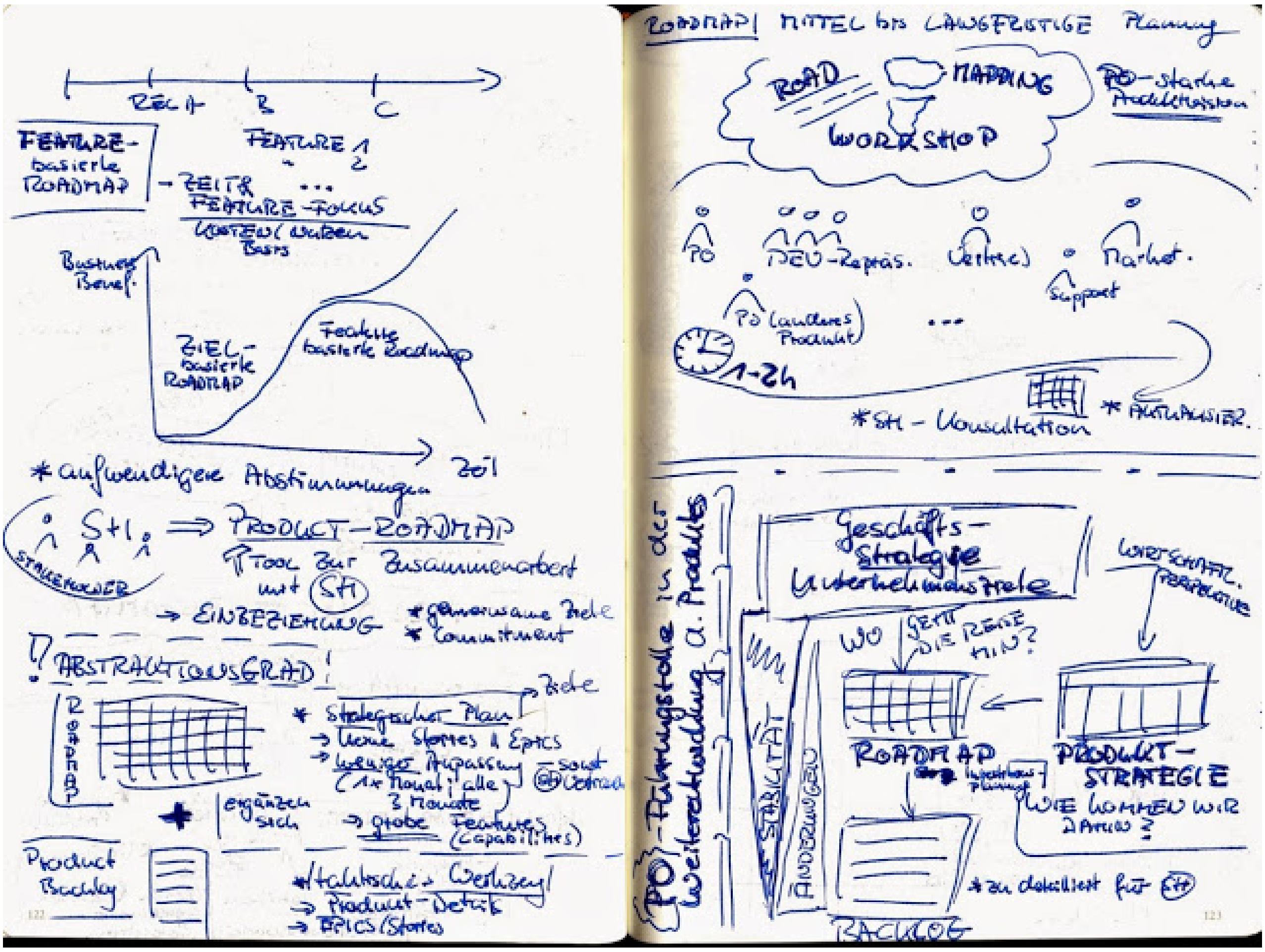
Benefits are less coordination and smaller backlogs. Disadvantages can be cross cutting aspects like UI and security that can be challenged by feature drifts.

Consider the Stacey Matrix for your product. Usually products flow from the chaos/complex towards the complicated/simple areas during their product life cycle.

Product roadmaps build the high level picture.

You can build goal oriented roadmaps structured by milestones/releases where goals or parts of your goals are achieved.

Goals can imply – learning and validation, engagement, retention, competitive advantages, market feedback, acquisition.

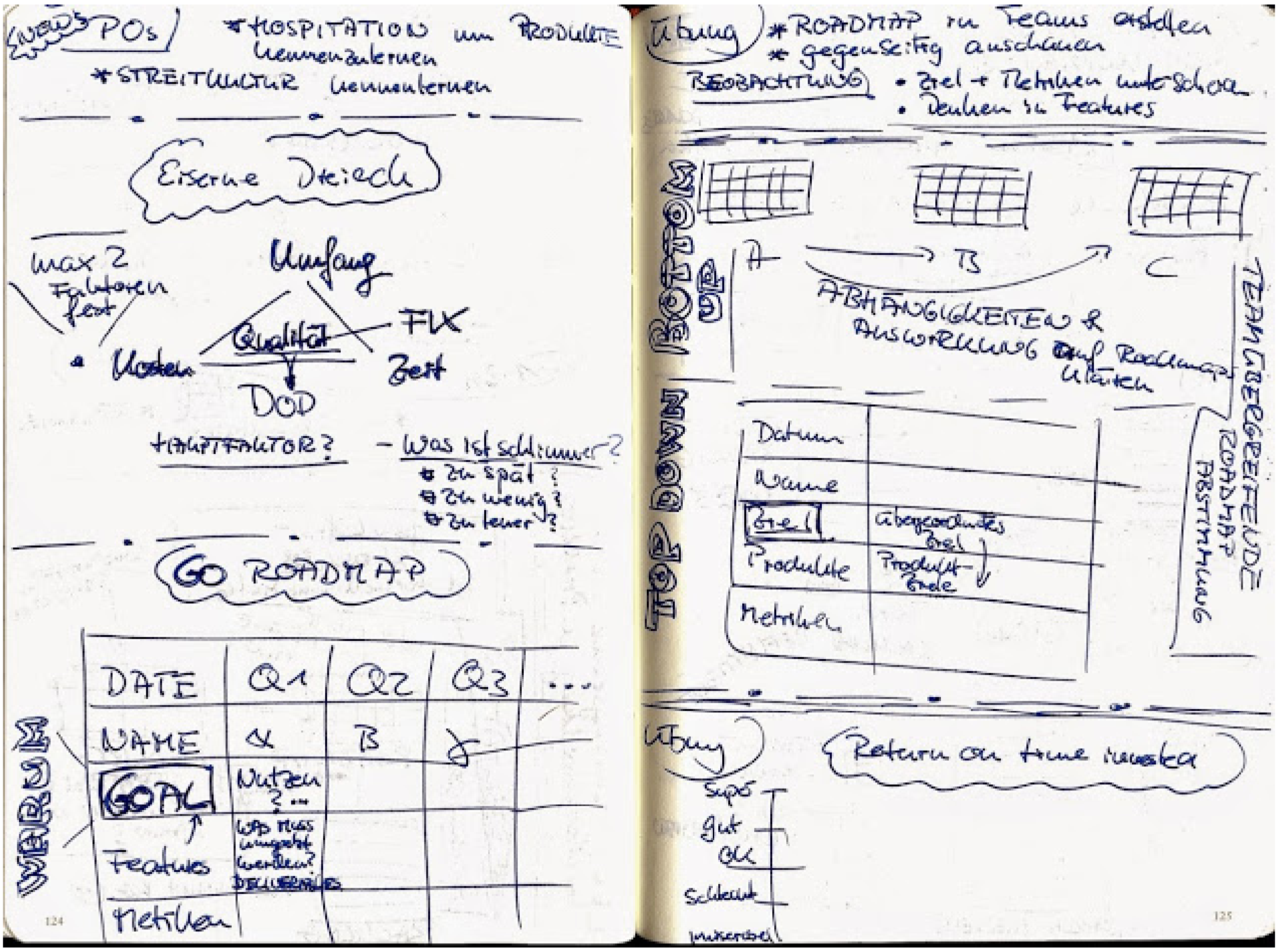


Another way to build roadmaps are feature based roadmaps. You map your features on the timeline.

Roadmaps

- are a tool for collaboration with your stakeholders
- are situated on strategical level
- should not consider epics and stories but features (higher abstraction level)
- are less frequently adopted than the product backlog (aim for once per quarter with all stakeholders)
- use a road mapping workshops (working with PO, Dev, Marketing, Support, Sales, …)

Company goals and business goals feed the product roadmap.
Economical data is input for the product strategy and influences the product roadmap too.
Based on the product roadmap the backlog is feed.



NEWS POs

* HOSPITATION um PRODUKTE kennenzulernen
* STREITKULTUR kennenlernen

Eiserne Dreieck

max 2 Faktoren fest

Umfang

Kosten

Zeit

Qualität — FIX

DOD

HAUPTFAKTOR?

– Was ist schlimmer?
- zu spät?
- zu wenig?
- zu teuer?

GO ROADMAP

WARUM

DATE	Q1	Q2	Q3	...
NAME	α	β	γ	
GOAL	Nutzen ?...			
Features	Was muss umgesetzt werden? DELIVERABLES			
Metriken				

124

Übung

* ROADMAP zu Teams erstellen
* gegenseitig anschauen

BEOBACHTUNG

- Ziel + Metriken unterscheiden
- Denken in Features

BOTTOM UP

A → B → C

ABHÄNGIGKEITEN & AUSWIRKUNG auf Roadmaps klären

TOP DOWN

Datum	
Name	
Ziel	übergeordnetes Ziel
Produkte	Produkt-Ziele
Metriken	

TEAMÜBERGREIFENDE ROADMAP ABSTIMMUNG

Übung

Return on time invested

super

gut

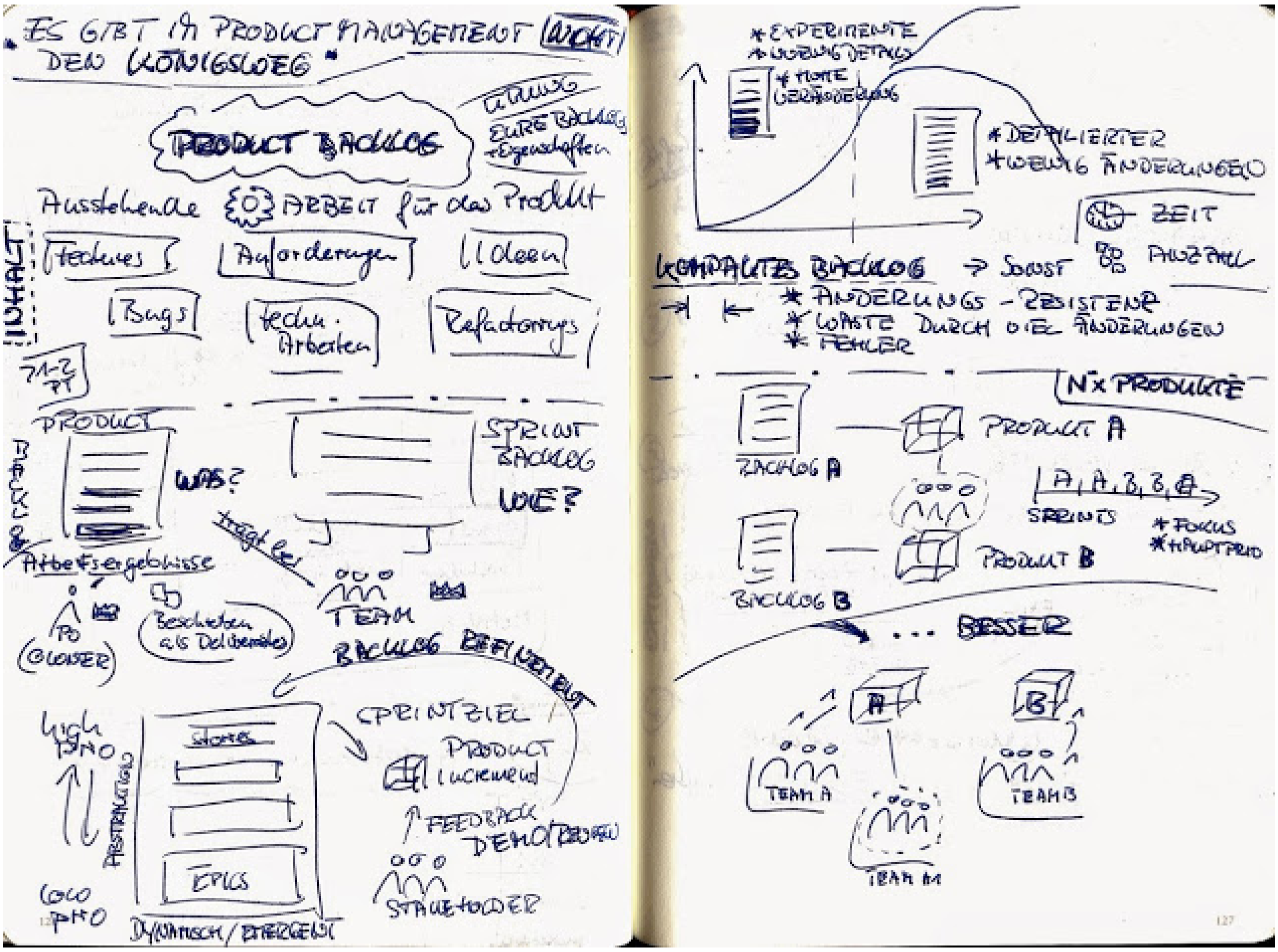
ok

schlecht

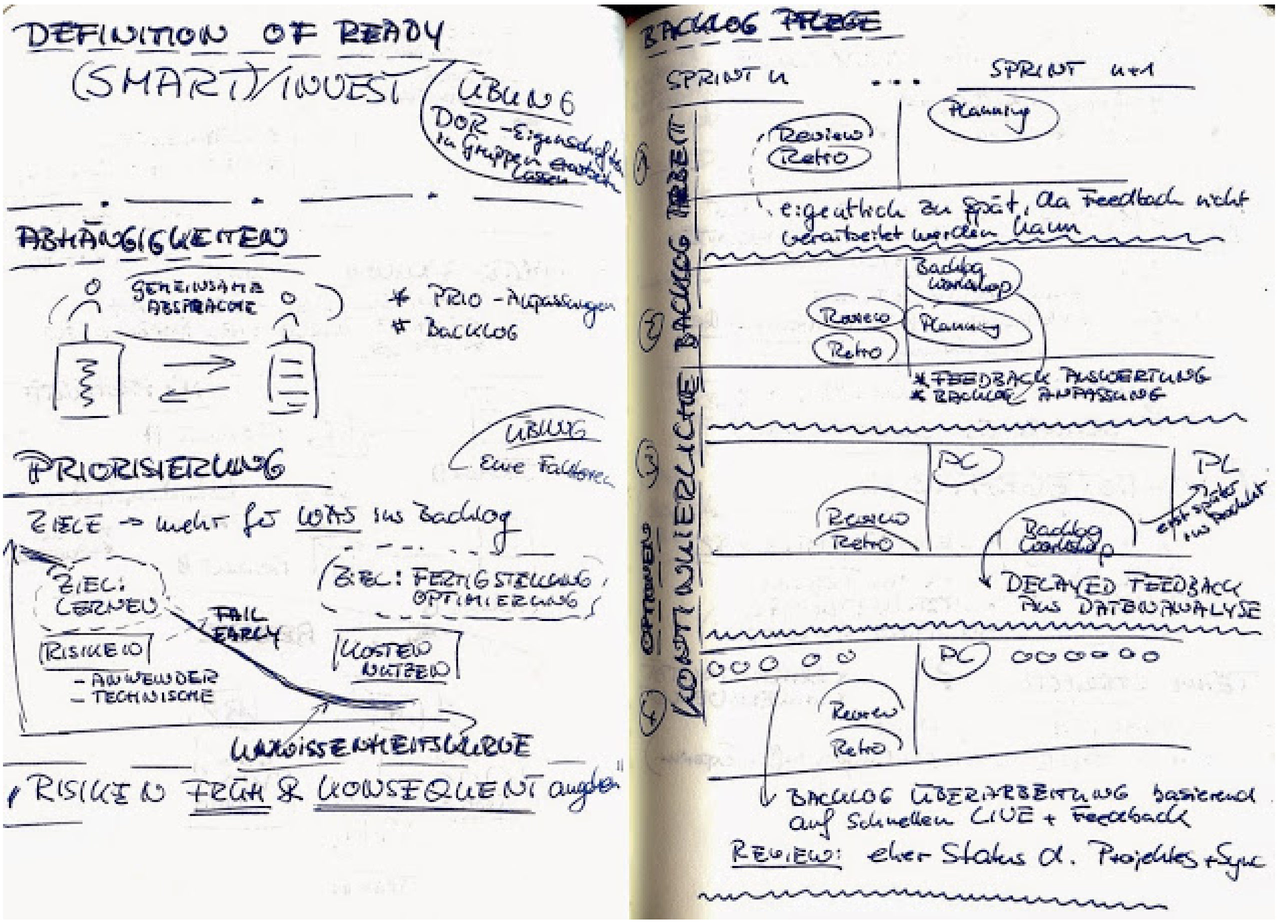
miserabel

125

Use hospitation for new product owners to learn about the products in your company.
Consider the magic triangle – costs, scope, time (and fixed quality). Only 2 of 3 can get fixed.
Use the nice roadmap template and put the goal (ask WHY) in the middle to derive features and metrics.
You can manage roadmap dependencies by checking all roadmaps and move parts in roadmaps accordingly. (Bottom up approach).
Using the top down approach you derive product roadmaps from a common goal.



There is now silver bullet/best way to apply product management.
The product backlog contains the WHAT to build, the sprint backlog the HOW to build it.
Use the sprint review to gather feedback from your stakeholders how to continue with your product development.
Consider the product life cycle when structuring your backlog.
In the experimental, high growth area consider having compact backlogs with high change awareness and adaptability – less details and more experiments.
The bigger the backlog the more you can trap to the tendency to avoid applying changes to the backlog.
Later on you can have backlogs with more details and expect fewer changes.
Use time and scope metrics to clean the backlog frequently.
It’s better to have multiple teams/a team building one product and pulling from one product backlog instead of having a team pulling from multiple backlogs.
If necessary at least minimize focus switching and focus on one product per sprint.



Use a definition of ready.

Use the curve of nescience for prioritization and match it to your product life cycle. Are you aiming for rapid learning or optimization and feature completeness?

Adress risks early and consequently.

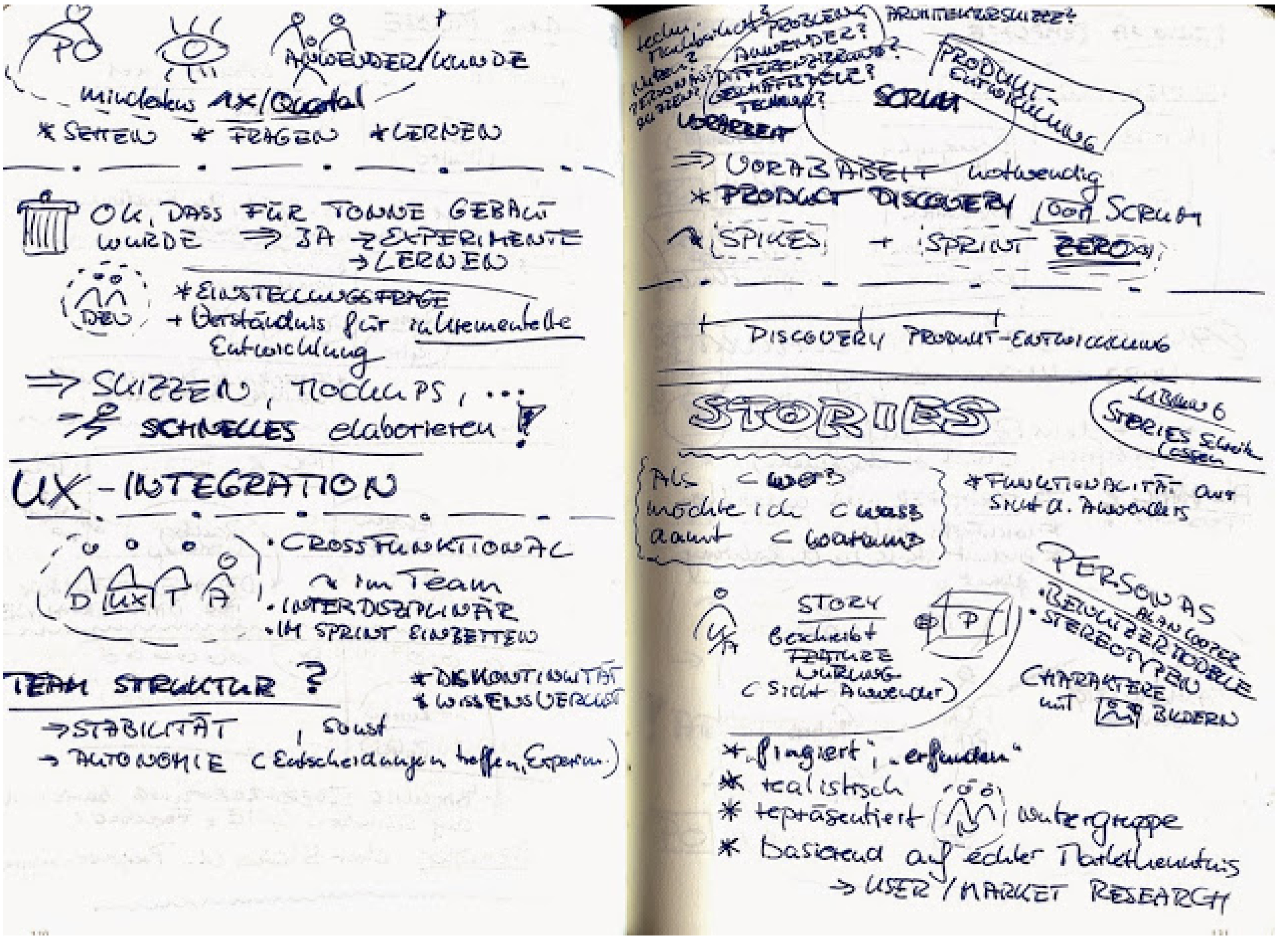
Consider 4 different ways for gathering feedback and product backlog refinement.

1) Traditional (in Scrum) short after sprint review and retrospective. Usually the time period is to short to proper adjust your backlog

2) Introduce a backlog workshop after review+retros to include feedback changes

3) Do the backlog workshop later on when product data is available.

4) Use instant backlog adjustments based on fast deployments and live software. Your review becomes more a status/project sync.

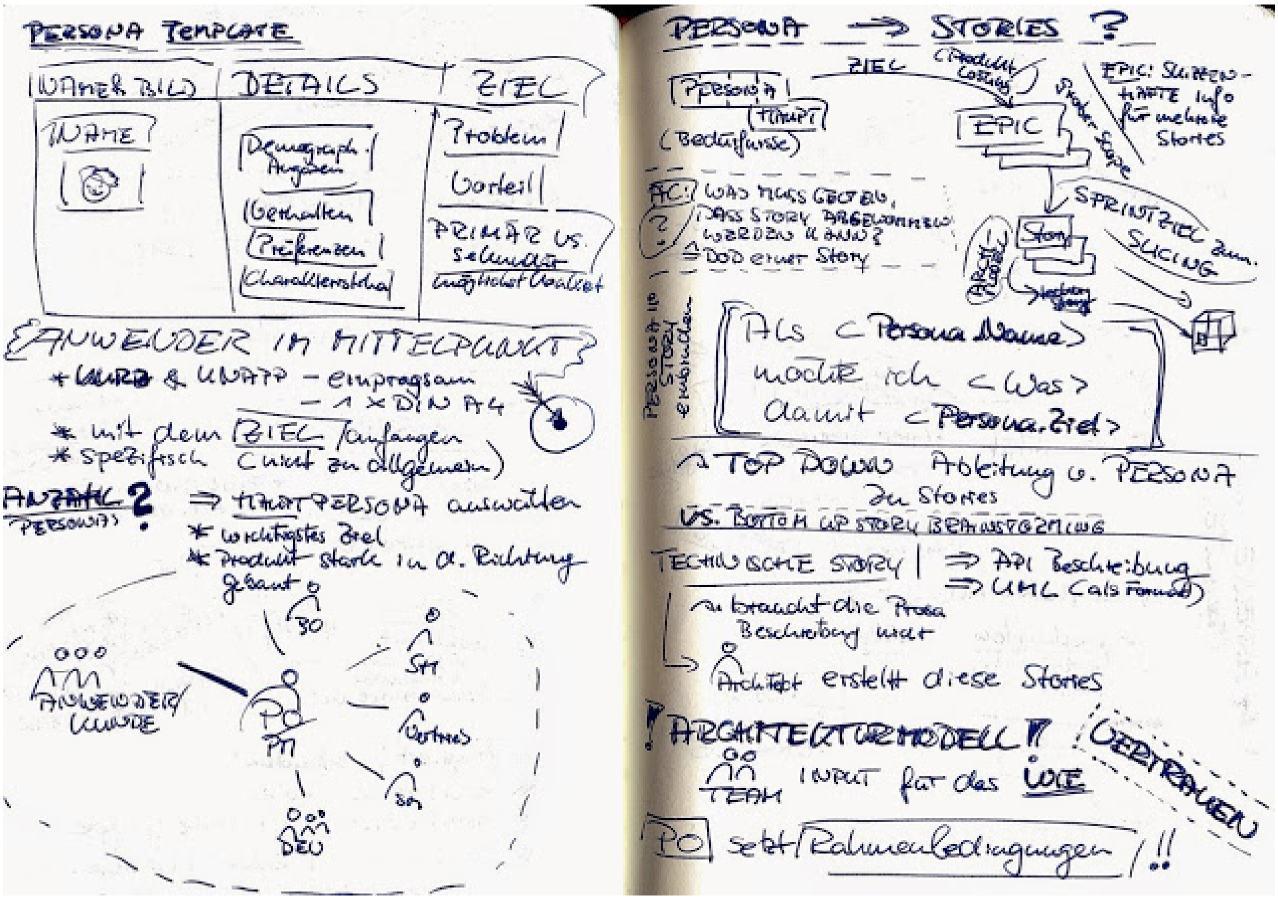


PO – meet your users at least once per quarter!
Embed UX in your teams and aim for team stability and autonomy.
Before starting with your Scrum environment proper initiate. Ask:

- What are your users?
- Needs/Goals/Roadmap?
- Architectural sketch…
- Personas
- Technology.

Apply Sprint 0's.
Use Personas – user models, user stereotypes:

- fictitious
- realistic
- represent a user group
- based on user/market research



Use the nice persona template mapping naming/picture, details and the goal. To the point using one A4 sheet, start with the goal.

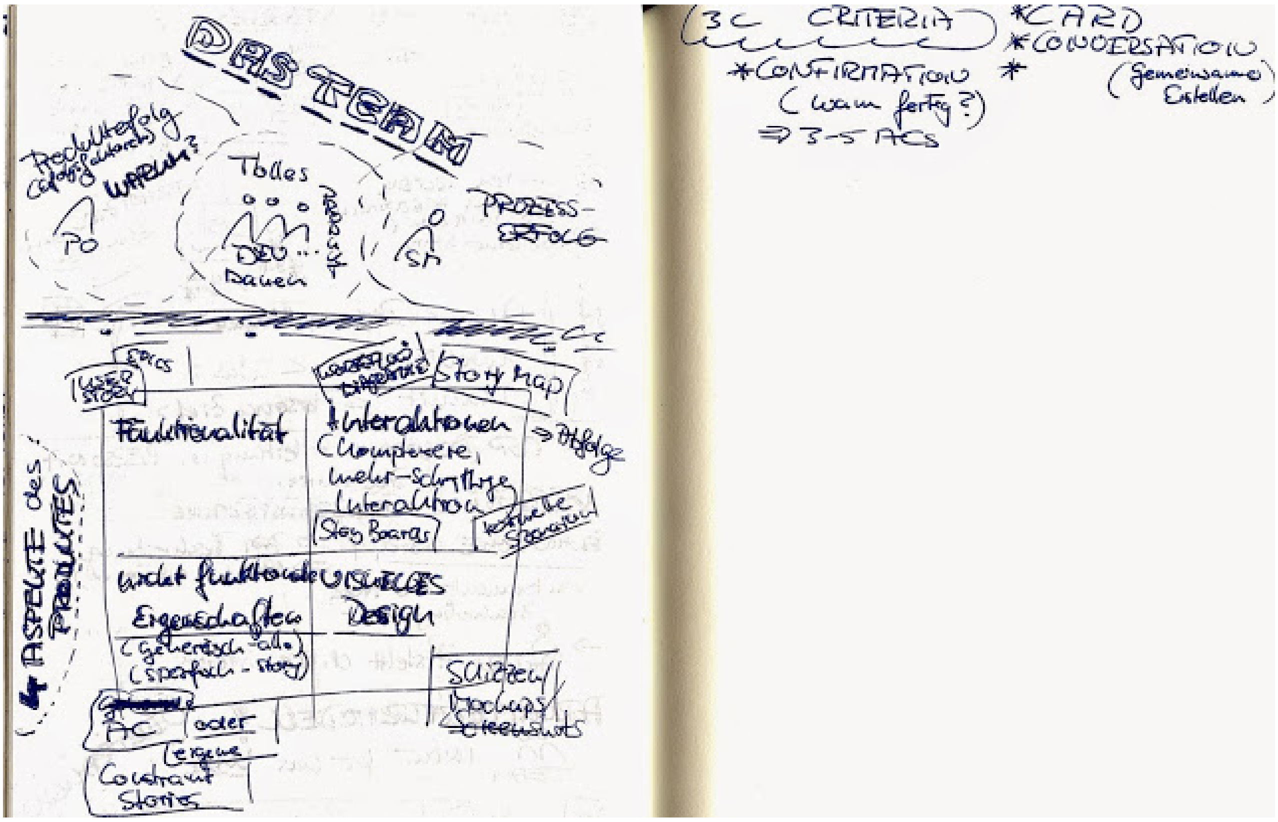
Use one main persona (oriented on the major goal). Use personas to drive your stories.

As <Persona.Name> I would like to <What> to achieve <Persona.Goal>.

Top down from persona to story.

Technical stories can by done via API descriptions or UML diagrams and do not need the story template format.

Should be created and maintained by the architect or technical knowledgable team member). Input by the architecture model.



PO – Why, Dev – Build the great product, ScrumMaster ensure process success.

Consider 4 product aspects:

- functionality (by user stories),
- non functional areas (acceptance criteria or constraint stories),
- interactions (story boards, story map, scenarios, workflow),
- visual designs (sketches, screenshots, mockups)

Thanks Roman for your great input, your patience and great answer to a lot of questions. It was really an experience!