

Why and how to avoid “product discovery” without teams?

(Why to avoid product discovery STAGE?)

Viktor Grgic @vgrgic, viktor@odd-e.com, Odd-e Netherlands B.V. odd-e.com less.works

Assumption of knowledge

We should know what we want

We should know how to build it

vs

We likely DON’T know what we need

We likely DON’T know how to build it



Not knowing —> system that maximizes killing bad ideas over delivering as many as possible

How to envision a product/capability TOGETHER (story about family banking)

Collaborative practices

Vision Statement	Crisp summary of the vision / idea.			
Target group	**Needs**	**Product**	**Value**	**Competition**
Which market segment does the product address? Who are the target users and customers?	Which needs does the product fulfil? How does it create value for its users? Which emotions will it evoke?	What are the three to five top features? What are its unique selling points?	How is the product going to benefit the company? Will it, for instance, increase revenue, enter a new market, develop the brand, reduce cost, create valuable knowledge?	Who are product's main competitors? How does it differ from them? What are its strengths and weaknesses in comparison?

http://www.romanpichler.com/ CC BY SA

Vision Statement	*A simple, fun, and affordable mega product.*		
Target group	**Needs**	**Solution**	**Value**
			

Planguage (Tom Gilb)

Tag	*A unique identifier of the quality requirement*
Gist	*Identifier a bit more explained*
Scale	*The scale of measure used to quantify the statement*
Meter	*The process or device used to establish location on a Scale*
Must	*The minimum level required to avoid failure*
Plan	*Realistic level considering we have limited budget and time*
Wish	*User and business would say “Wauw!” if we achieved this level*
Current	***Currently measured level after each sprint***

Planguage (Tom Gilb)

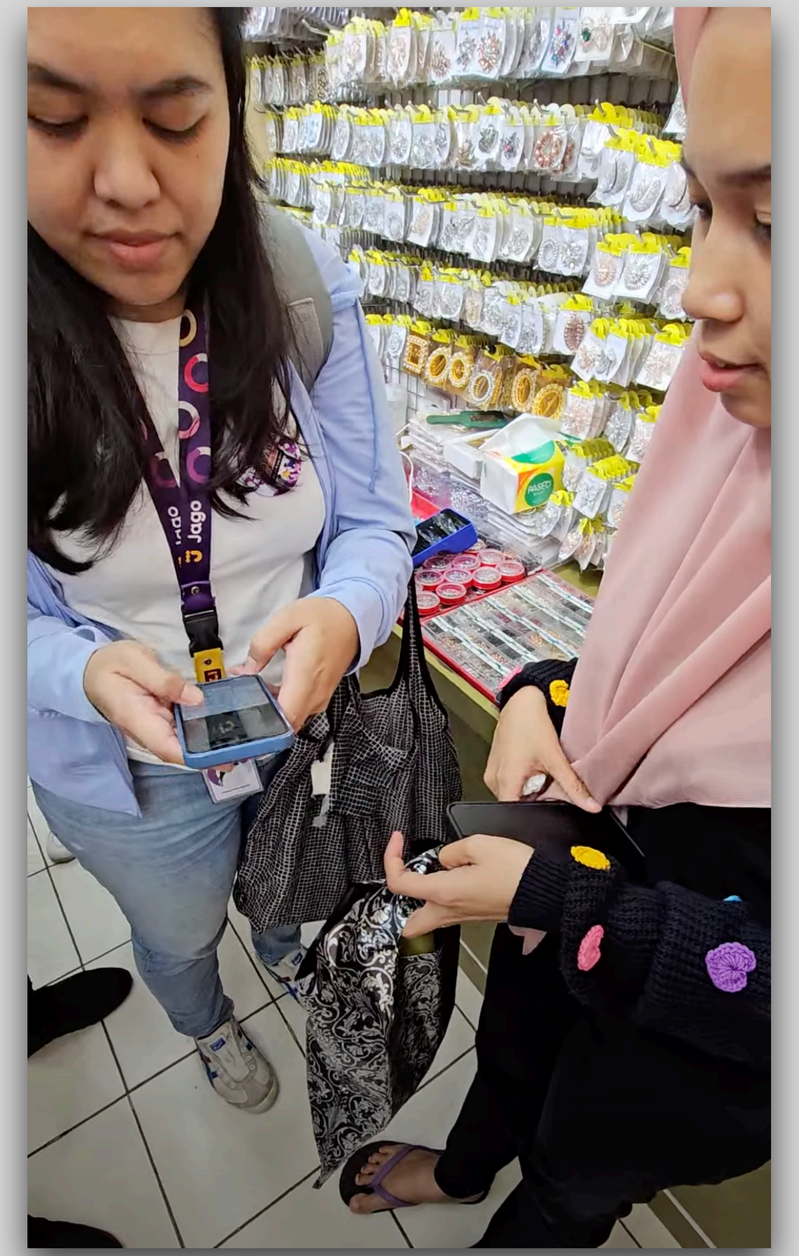
Tag	*Learnable*
Gist	*The ease of learning to use the system*
Scale	*Time required for a Novice to complete a 1-item order using only the online help system for assistance.*
Meter	*Measurements obtained on 100 Novices during user interface testing*
Must	*No more than 7 minutes 80% of the time*
Plan	*No more than 5 minutes 80% of the time*
Wish	*No more than 3 minutes 100% of the time*

Current 15 minutes 80% of the time

Product Backlog

Complete teams together with subject matter experts

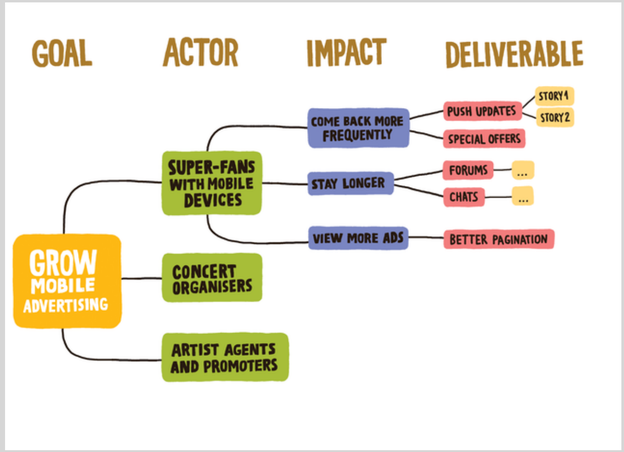




Ideation day

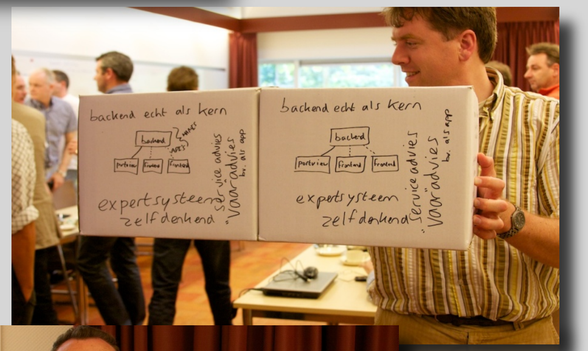
Collaborative practices















Maximize killing bad ideas





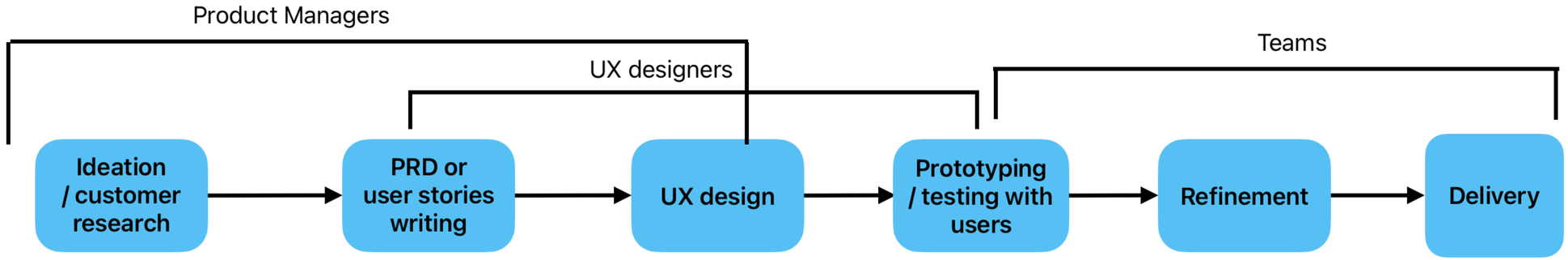
Individuals and interactions over processes and tools

should lead to

parallel over sequential

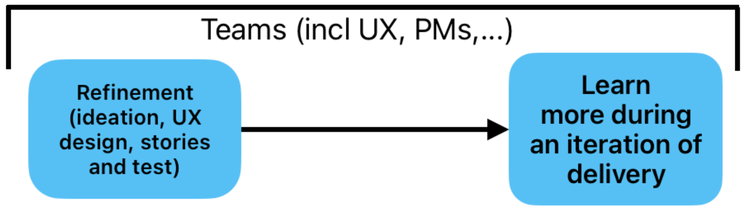
Sequential vs. parallel “discovery and delivery”

Avoid this





Try this



Step 1: Revisit the context, objective(s)
Step 2: Dive into details of the next step

Adapt plan daily based on learning



Asking for feedback is NOT collaboration

Problem with prototyping

- Assumption that defining without developers and testing a prototype gives sufficient knowledge to “deliver” a feature
- An idea can only truly be tested when it is technically delivered
- Misplaced perception from stakeholders that we are almost there (“just need to build it”)
- Avoid: prototype approvals
- Try: design prototype together with team and treat as just an artifact of delivery
- Try: favor a small product increment that addresses also technical risks over a prototype

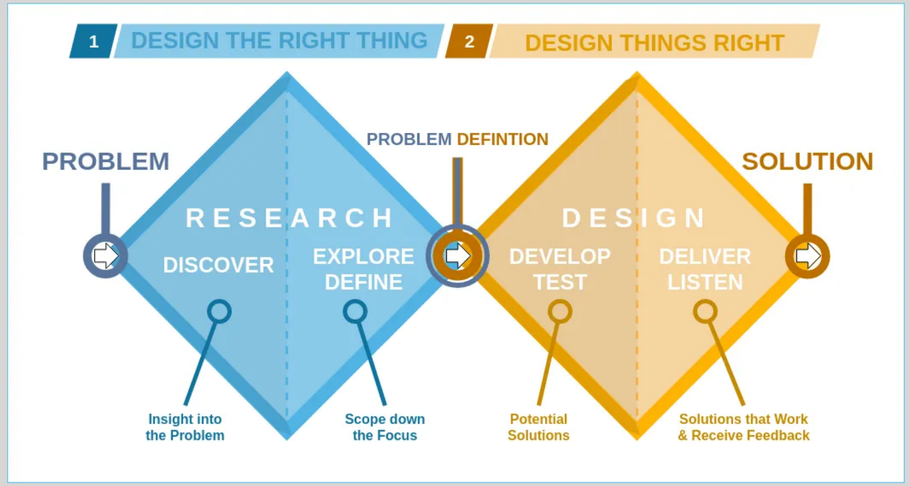
How do UX designers work with others?

- Assumption: “Designers cannot be in teams since there are **not enough** of them”
- Assumption: “Designers should be **outside** of teams since they are looking at the bigger picture and not single feature.
- In LeSS, having designers outside of teams leads to a painful and confusing situation
- Try: wireframing **together** with other team members
- Try: work on details only after product idea is **first discussed in refinement** with teams
- Try: other team members work on design, and UX designer is a **mentor**.
- Avoid: using FIGMA and other fancy tools to **collaborate** on design

Statements and concepts that prevent parallel work

“Most product people today do seem to understand the concept that in product discovery we’re “building to learn” and in product delivery we’re “**building to earn.**”

AVOID



"Once we have discovered a solution worth building, then we can proceed to build and deliver a production-quality solution (one that is reliable, scalable, maintainable, performant, and secure).”

“Once we have the necessary evidence that we have discovered a solution worth building, we now need to **communicate to** the engineers the many details of the necessary product.”


EVERYONE
EVERYTHING
EVERYWHERE
ALL AT ONCE
天馬行空
COMING
SOON