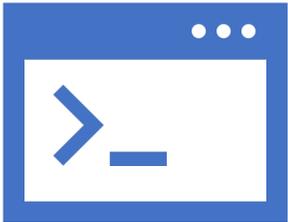


Experience

# "Phasenangemessenheit"

...phase adequacy.

...or to be actually precise:



**"Entwicklungsphasenangemessenheit"**

# Disclaimer

All presented opinions are personal opinions of Frederic and Konstantin and do not express the views or opinions of their employer.



# Content

- What are phases?
- What is "Phasenangemessenheit"?
- Why is "Phasenangemessenheit" a thing?
- Can the benefits be achieved easier?
- Anti-patterns to look out for
- How does the release-to-customer cycle-time influence the phases?

# What people learn from this session?

1

**Understand** the forces at play when it comes to phases and the resulting "adaptations" of structures and process.

2

**Learn** how to detect misunderstandings or hindering processes that may support phases.

3

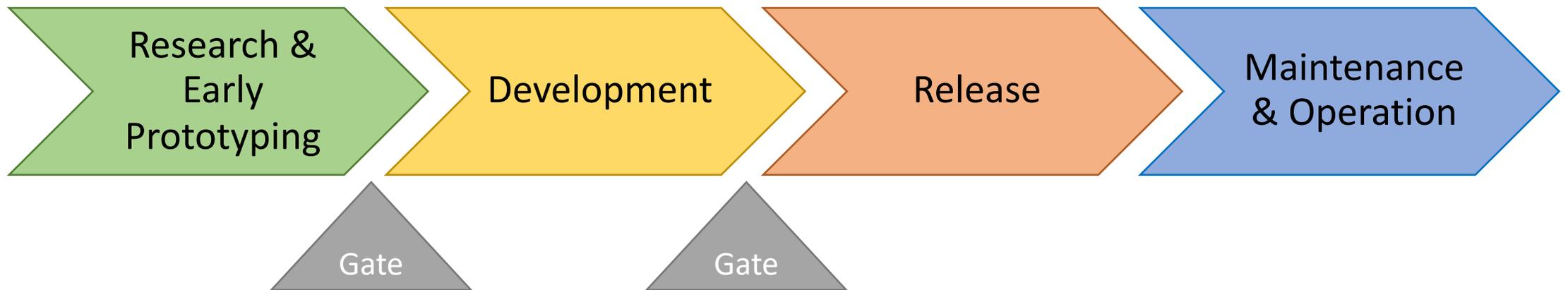
**Understand** the argumentation why phases exist and learn about possible counter arguments.

# What exactly are phases?

## Based on our experiences, they...

- ⦿ ...describe the implementation of a product development process in a company.
- ⦿ ...are typically derived from a process model, e.g., BAH or stage-gate.
- ⦿ ...are targeted to facilitate or smoothen the processes of the company.
- ⦿ ...rather describe a timeline (Just before launch) or the expected environment for teams (high pressure phase, Death March\*).

# What exactly are phases? Example:

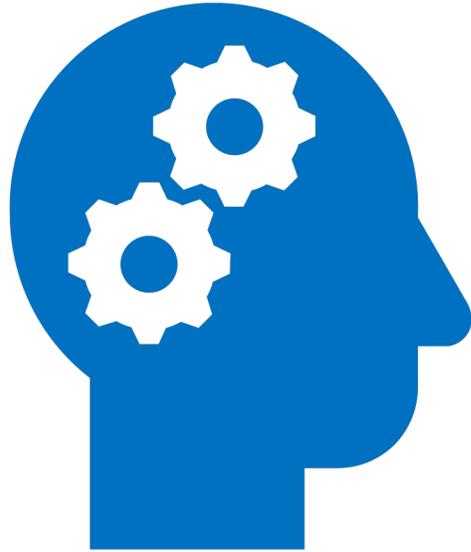


High risk for phases to become a sequential life-cycle.

# What is "Phasen- angemessenheit"?

Based on our experiences, it...

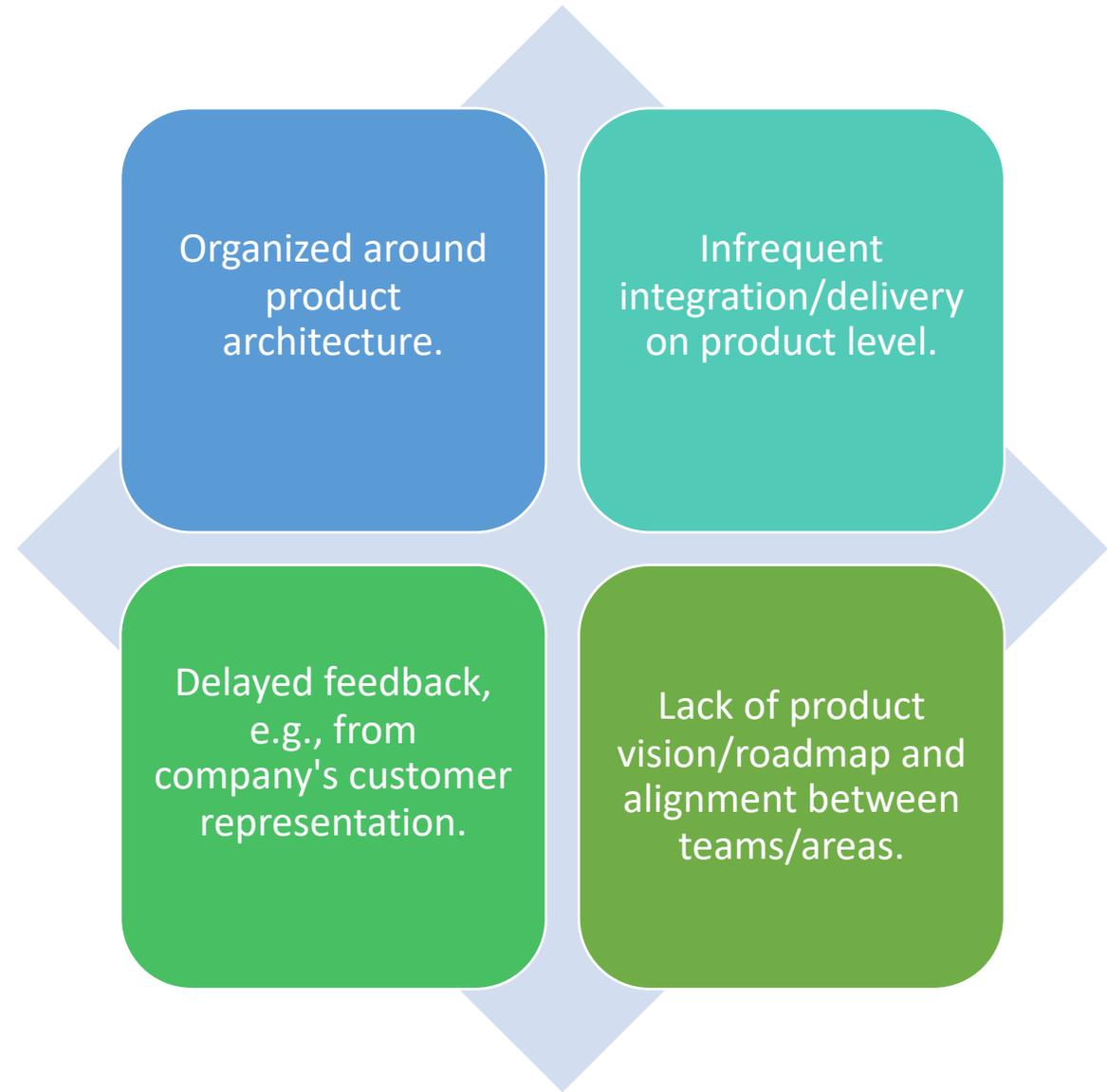
- ⊙ ...assesses whether a method/tool/action/process is applicable in the current *phase*.
- ⊙ ...is used as *ratification* to start/stop doing events.
- ⊙ ...is understood by the management as means to *increase efficiency*.
- ⊙ ...does not increase efficiency & *hinders learning*.
- ⊙ ...piles up *technical debt*.



## “Phasenangemessenheit” Why is it a thing?

- ⦿ Phases are perceived as integral part of the product. Hence, they cannot be changed.
- ⦿ Positive effect: It reduces processes and gets things done, especially in large organizations.
- ⦿ Keeps process departments busy and fosters heroism (*"I saved the process & the product"*).
- ⦿ Easy to understand (*"But we have to deliver!"*) without deeper understanding of underlying processes & technologies.

# Anti- patterns to look out for



# How does the release-to-customer cycle-time influence the phases?

01

Motivation to reduce the cycle on company-level.

02

Joint integration on product level.

03

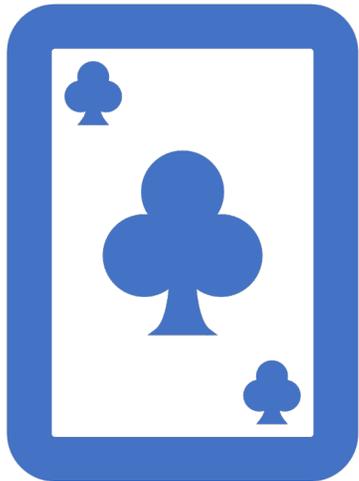
Improve ROI of customer offerings.

# Can the benefits of phases be achieved easier?

- ⦿ Decrease cadence, check progress and make go/kill decisions *every* iteration.
- ⦿ Company organized around customer-centric requirements
- ⦿ Adaptive technology stack & foster (code) reuse
- ⦿ ...

# How about a game?

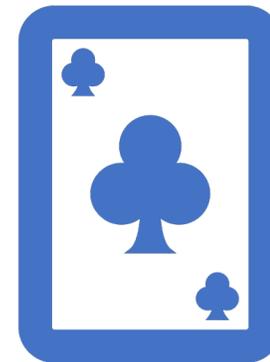
---



- ⦿ **Anticipation Game:**  
"Think about your context in your company. What are arguments that would come up to manifest that phases are inevitable?"
- ⦿ Write down counter-arguments!
- ⦿ We will present arguments and you can only use your predefined answers to challenge them.
- ⦿ "Best" answer wins...
- ⦿ Team BLUE vs. Team PURPLE

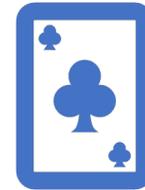
# The Rules

- ⦿ You have to anticipate our challenges.
- ⦿ At maximum 15 words per card.
- ⦿ Maximum of 5 cards.
- ⦿ No reuse of cards.
- ⦿ You are **not** forced to play a card.
- ⦿ The leading team goes first in every round.
- ⦿ One Bonus-Round in case of a tied game.



# Selecting the "best" answer

- ⦿ The more concrete your answer, the better.
- ⦿ Practical use of LeSS principles.
- ⦿ Propose possible experiments. 🧪
- ⦿ Embed your answer in the real world/your context.
- ⦿ Konstantin & Frederic are the judges.



## Round 1

Our research topics need flexibility and can not be bothered with industrialization requirements, otherwise the teams are too slow. Hence, we need the early phase to actually find out if the research topics can be put into the product.

## Round 2

The speed of change in the late phase is much quicker than one Sprint, hence, we can not wait for planning/refactoring events to happen. We need the late phase for that.

## Round 3

The only time when we can properly plan ahead is after the early phase, before the delivery pressure hits.

The events and all that were actually designed for that phase. We can stick to them only in that phase.

## Round 4

All the bugs from our **BUGTRACKING TOOL** have to be solved immediately. The amount of reported bugs will rise in the late phase, hence, we have to ignore the planning stuff. Otherwise we can not solve the problems in time.

## Round 5

Without the phases we are not able to handle the different projects and deliveries we are facing. This will help us to work more efficiently.

# Bonus Round

*The team that scored last has to answer first. The other team proposes a situation.*



Thank you for your  
attention!



Please give us some feedback!

*Go to...*

<https://www.menti.com/>

*...and use this code:*

**4983 1482**



*Scan me!*

*Go here!*

<https://www.menti.com/4ydfho42rz>