

Towards Continuous Everything

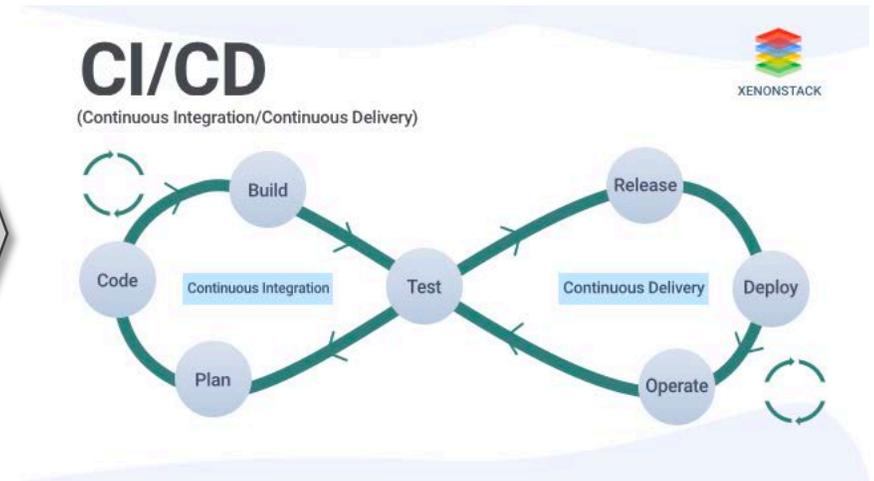
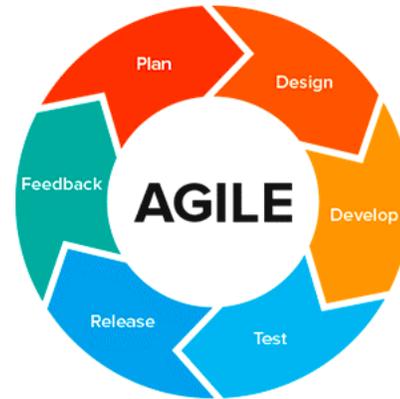
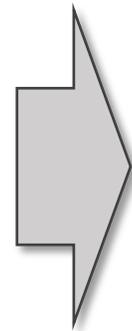
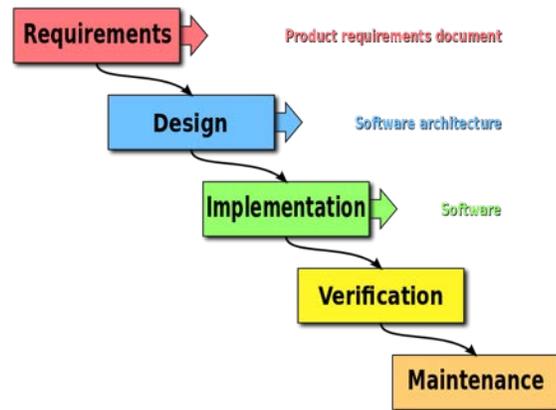
Eriks Klotins, BTH

BTH **SERL Sweden**
LEADING SOFTWARE ENGINEERING

S.E.R.T
reThink•reThought

A story about shipping

From Plan-driven, to Agile, to Continuous



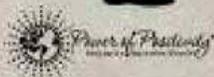
- Release time in years
- All project value (and risk) is delivered at the end
- In the worst case, it takes years to identify and correct a problem
- Relies on upfront process & planning

- Release every few weeks
- Value is delivered in chunks throughout the project
- It may take a few weeks to discover and fix a problem
- Relies on flexible collaboration

- Release immediately
- Value is delivered continuously in small increments
- Data enables rapid course adjustments
- Relies on automation, telemetry, and customer feedback

I fear not the man
who has practiced
10,000 kicks once,
but I fear the man
who has practiced one
kick 10,000 times.

- Bruce Lee

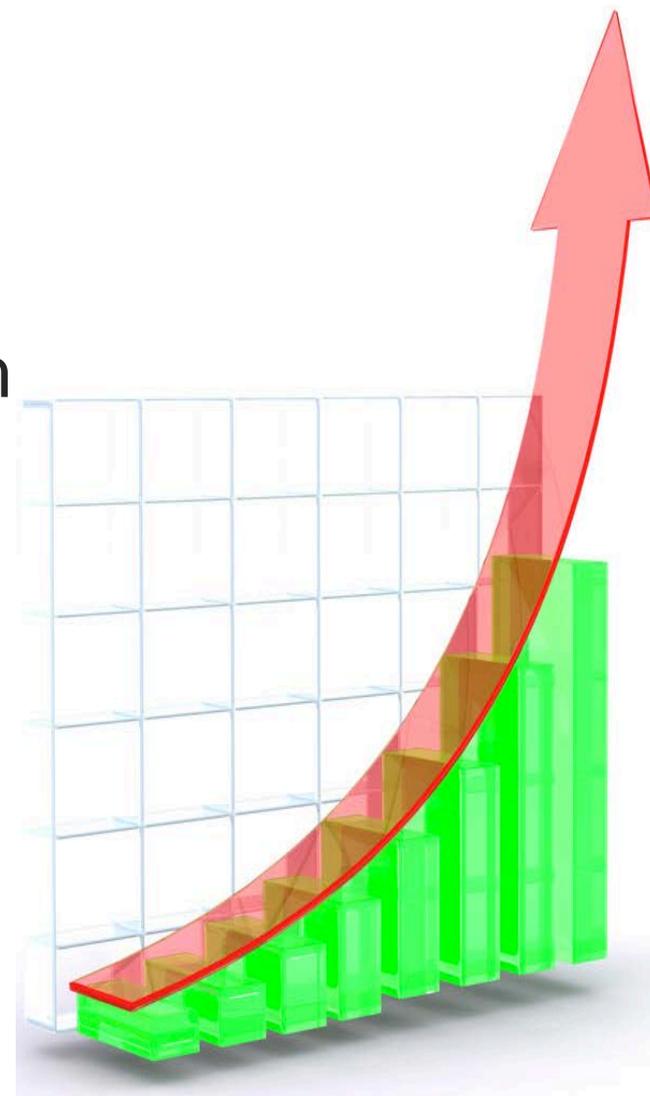


Two ways to look at speed

1. **Releasing 10 new features every month** will likely give you a bloated product and mediocre customer satisfaction
2. **Iterating and polishing one feature 10 times** every month will give you a chance to fine-tune the product to the exact customer needs

The Promise

1. **Effortless** and frequent product **updates**
2. By a magnitude **shorter feedback loop** with customers
3. More capacity for discovering new features with **experimentation**
4. Customer feedback & telemetry to support **data-driven decisions**
5. New **business** models and **value** streams
6. **Exponential advantage** in the market



The Assumptions

- The product is **suitable** for continuous deliveries
 - e.g. not mission/safety critical, modular, online
- **Customers want** frequent upgrades and are ready to share their (and their end-user) data
 - There are few downstream dependencies on other systems or processes
- **The whole organization** is ready for transformation
 - Once technology is there, marketing, sales, planning, etc. will follow
- **Automation** solves everything

The Reality

- **Retrofitting** an existing product with CI/CD is a **huge undertaking**
 - No one-solution-fits-all in rethinking existing processes, architecture, tools, and ways of working
- Automating sub-optimal processes is a **waste multiplication**
 - CI/CD is not doing the same just faster. It is a new way of doing development, delivery, QA, operations, business.
- **Customers** work on their own pace and **wish to plan** for product upgrades
 - Especially, if they have integrations and business dependencies
- Customer/end-user data are **very sensitive**
- Organizations are **cautious to tear down** their existing business models

The Consequences

- The main challenge is to **transform the organization** to work in a "continuous" way
- End-to-end CI/CD **may not be suitable** for everyone
- Organizations still can benefit from implementing **some CI/CD principles**
- Adopting CI/CD is not binary, it a **continuous process**

What we do

- What are the the **investments (costs)** and **benefits** of continuous software engineering?
- How to realize and **maximize** the benefits?
- How to structure the adoption and identify **low hanging fruits**?

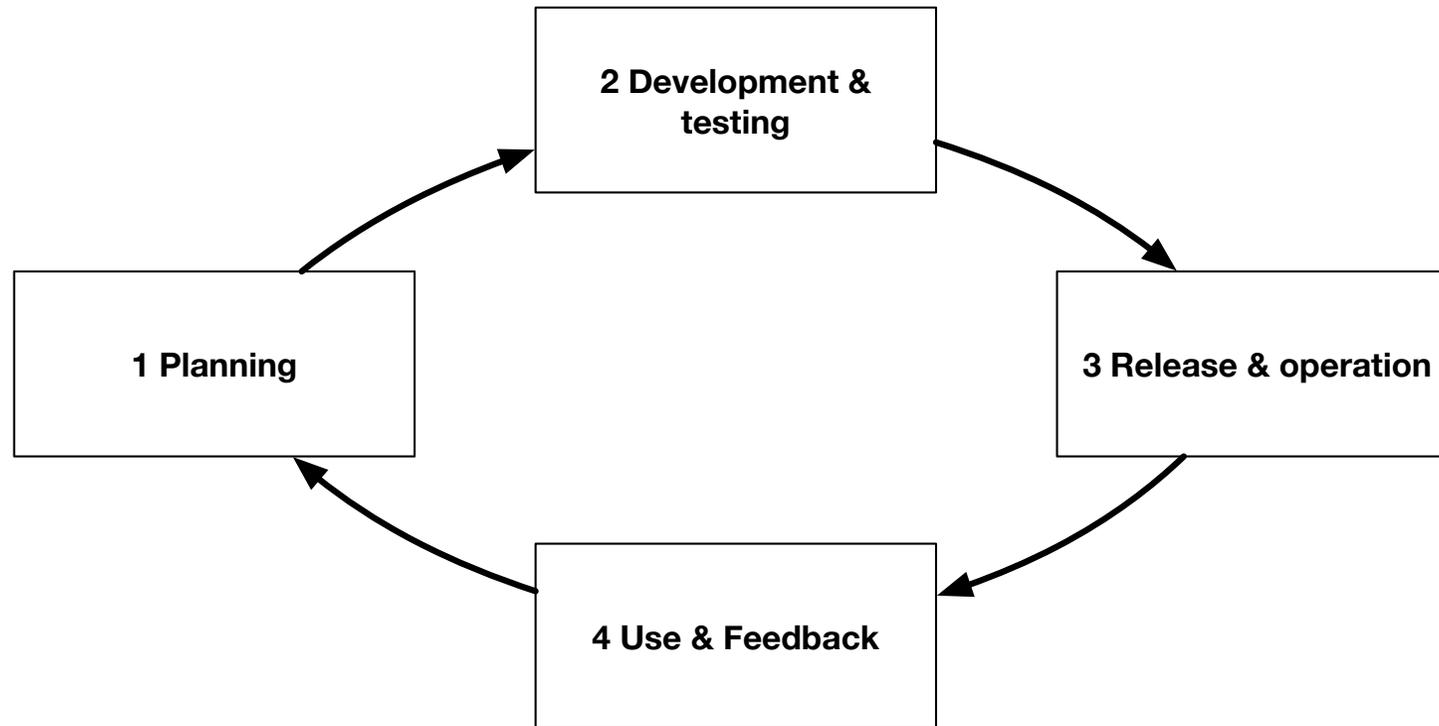




STATE OF THE ART

The Simplified Model

Visualizes the key areas
Present in every software organization (continuous or not)



Four organizations (levels) represent a typical setup of product stakeholders.

They can be different companies or parts of a single company if the product is aimed for internal users.

Level I: Engineering

Level II: Product

Level III: Operations

Level IV: End-user / customer

Level I: Engineering

1 Continuous
product planning

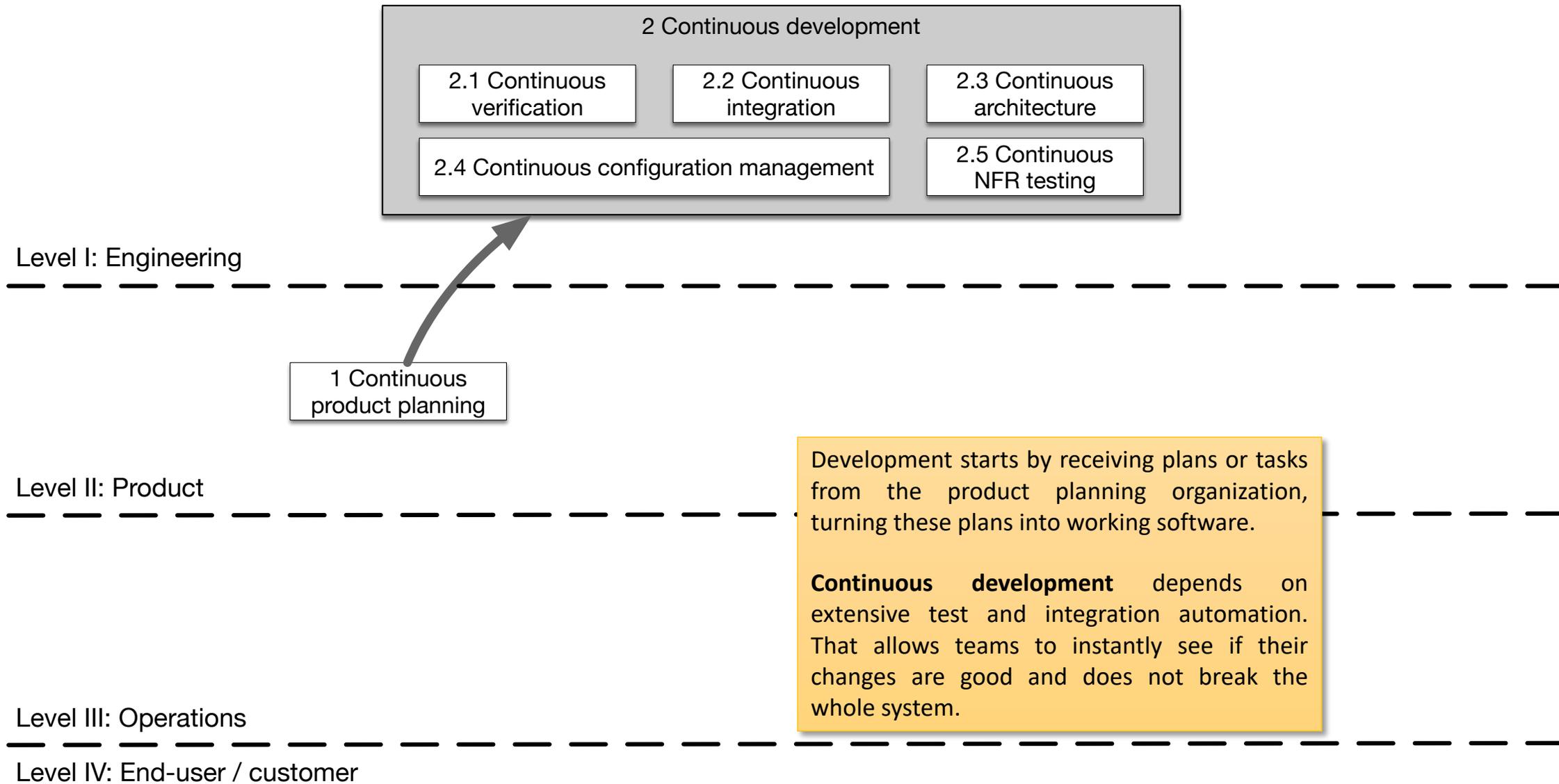
Level II: Product

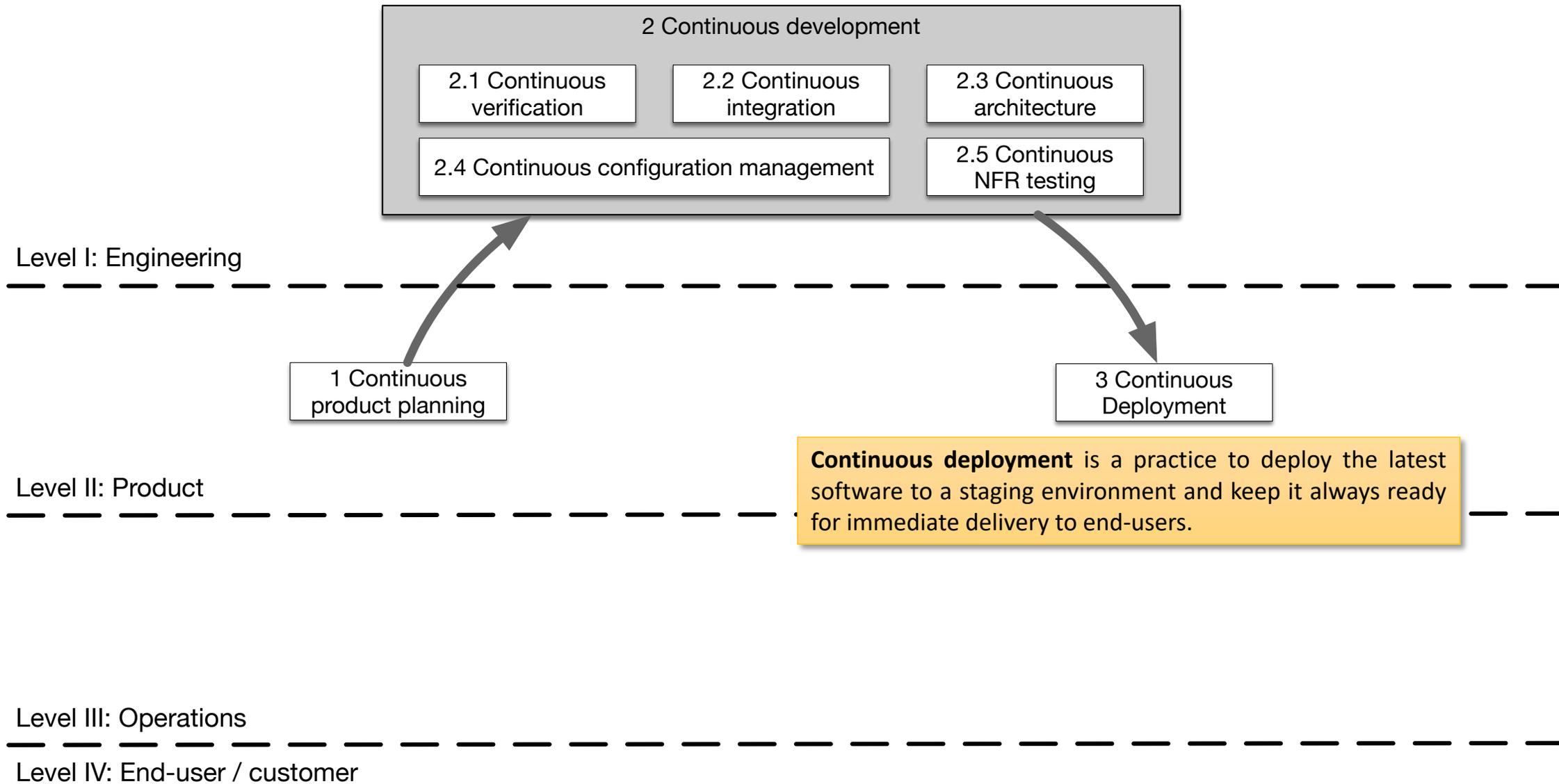
Planning is a holistic activity involving multiple stakeholders to create lightweight, dynamic, and open-ended product plans.

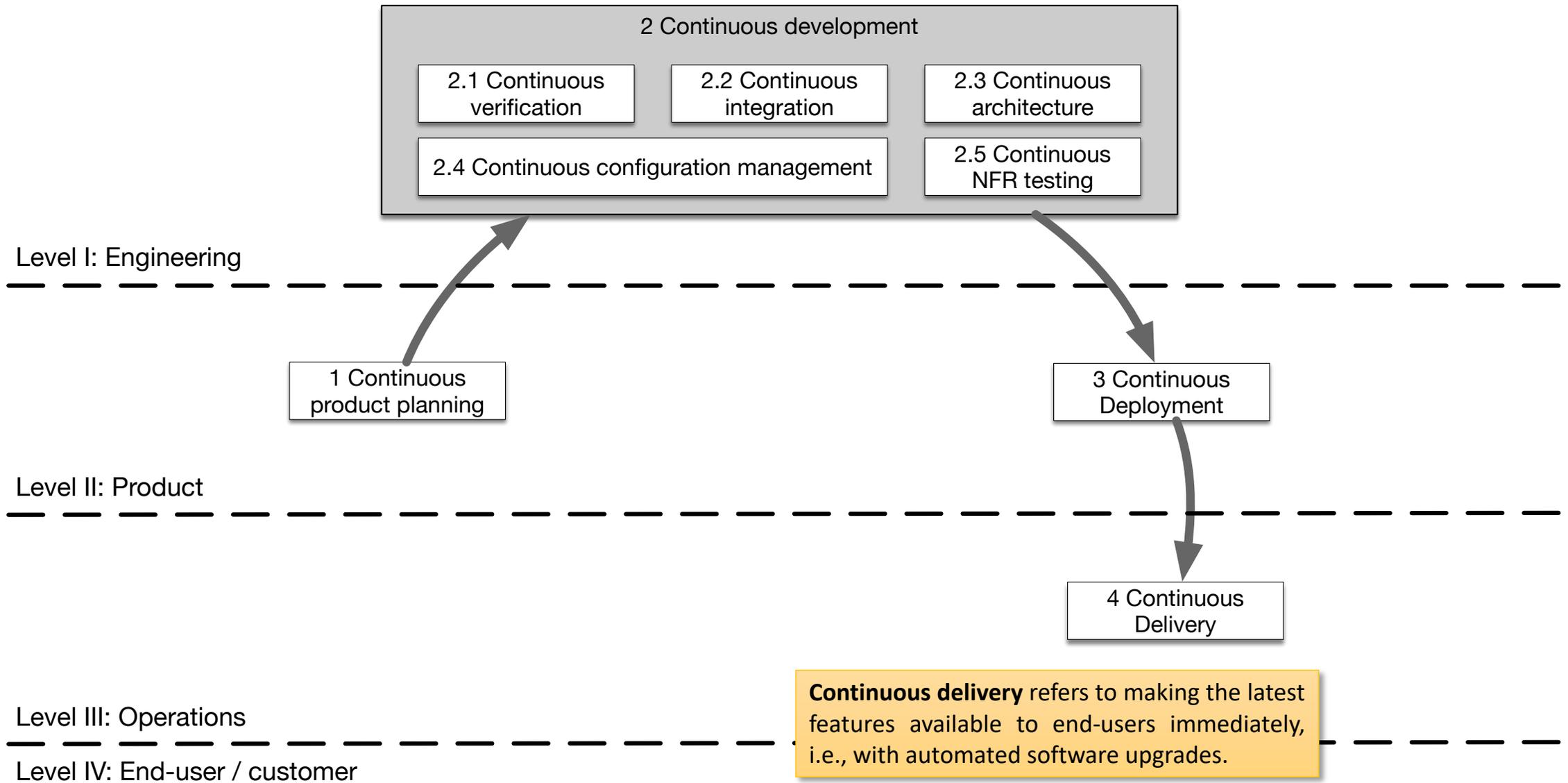
The planning is aimed to swiftly address and adjust to new market opportunities, changes in a business environment, and technologies

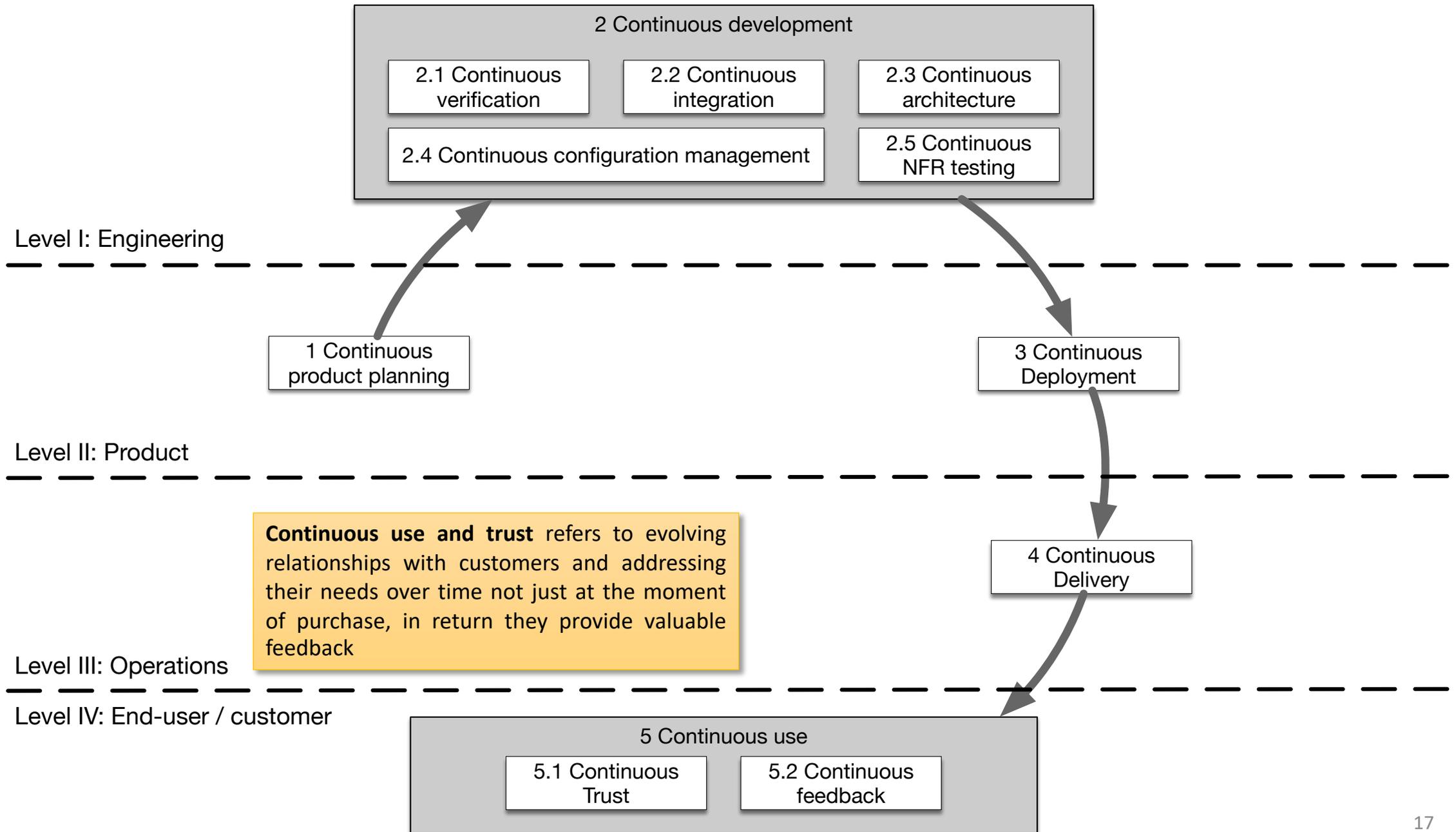
Level III: Operations

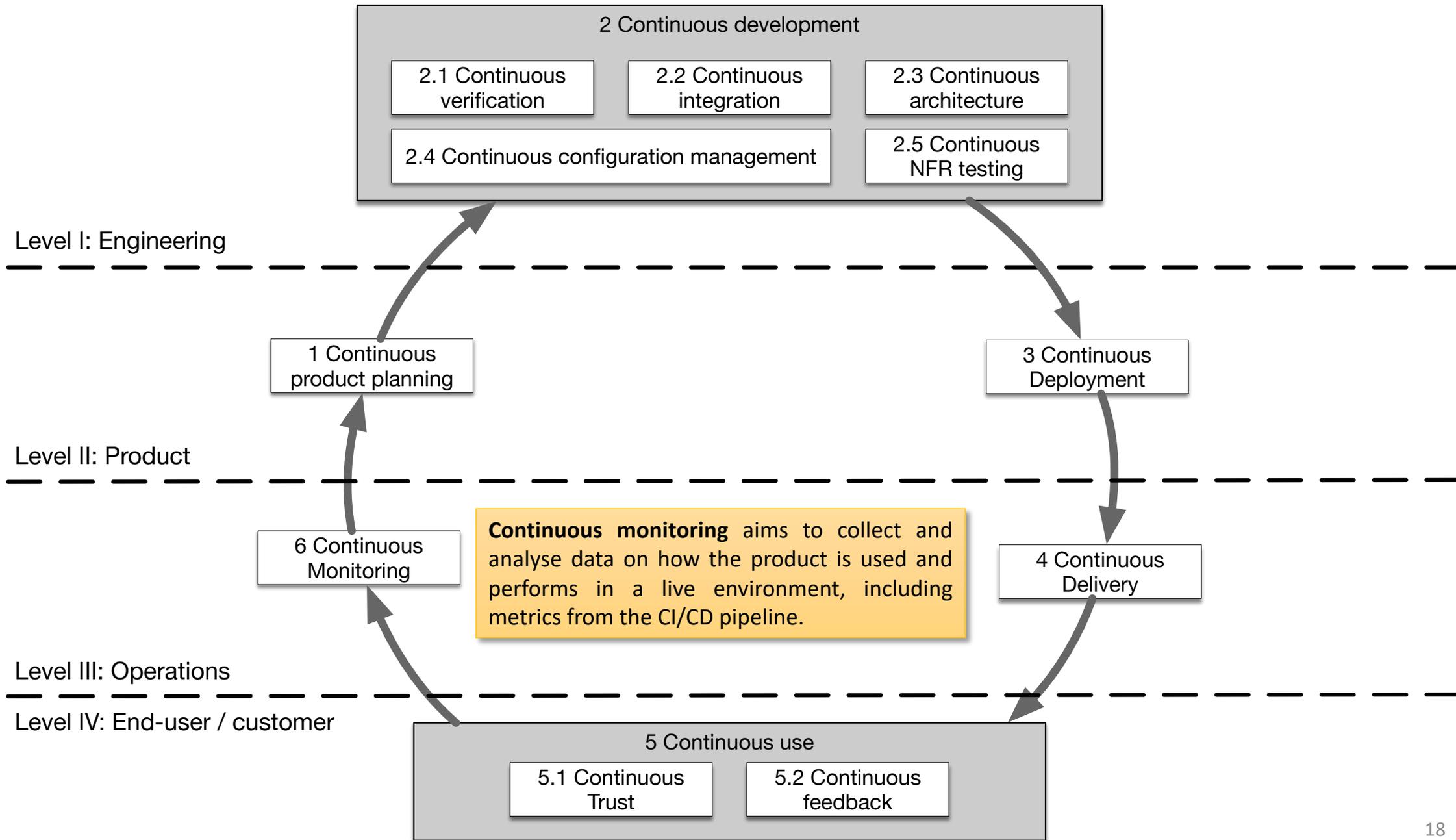
Level IV: End-user / customer

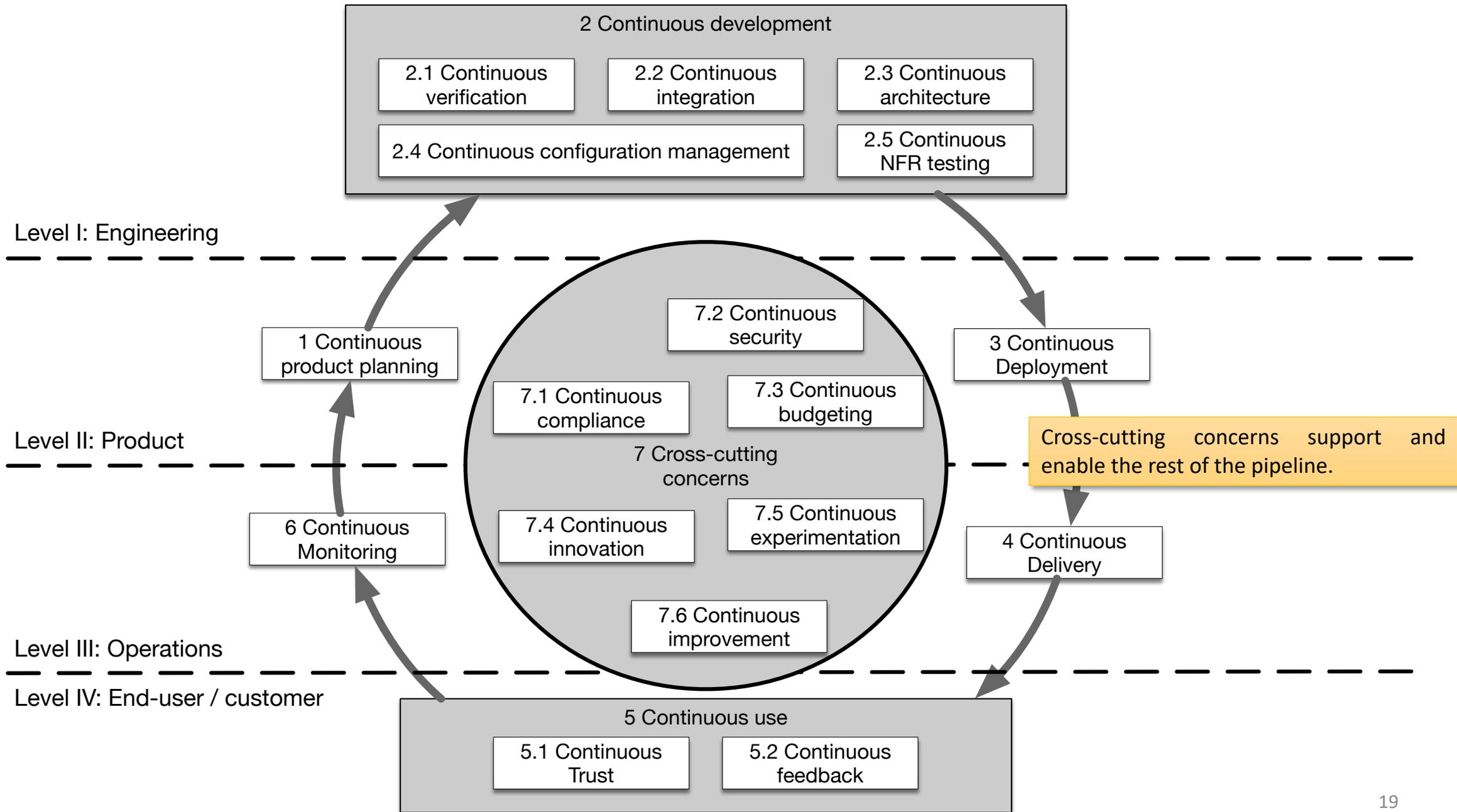






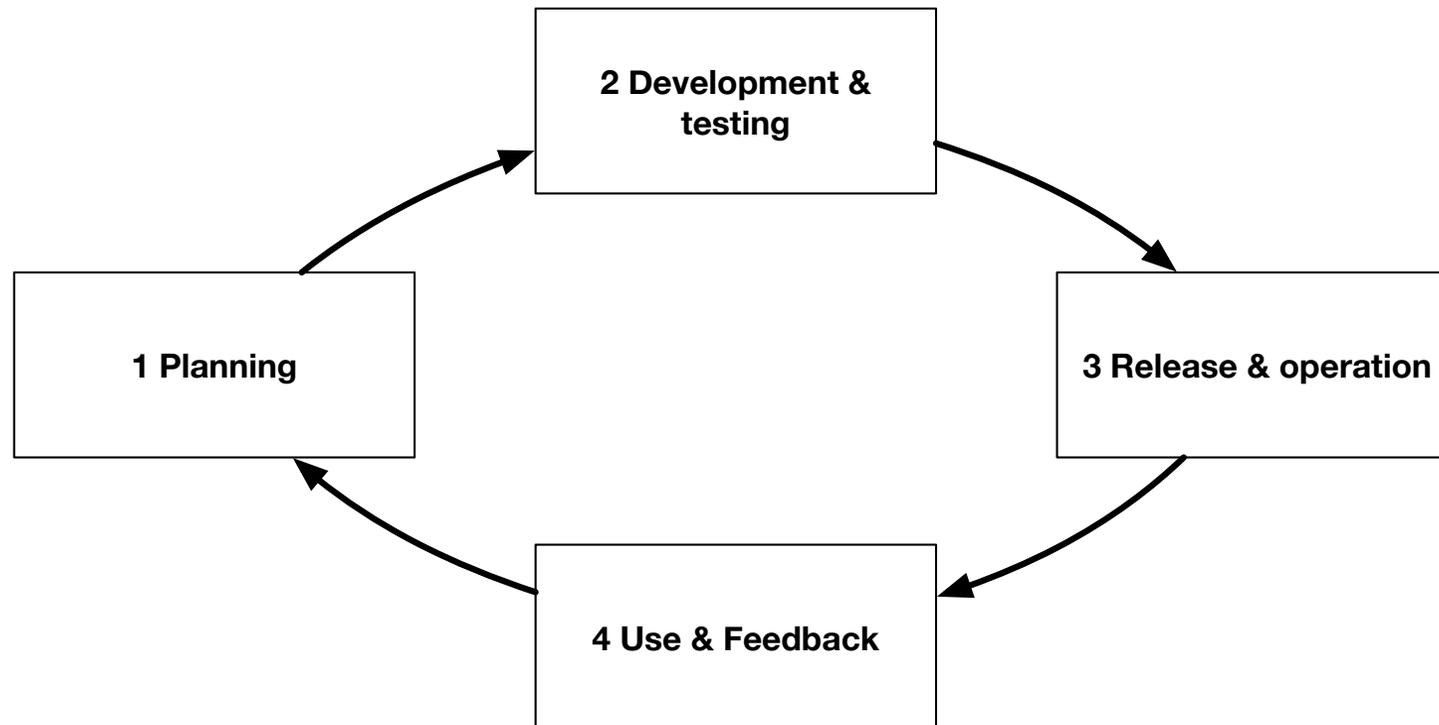




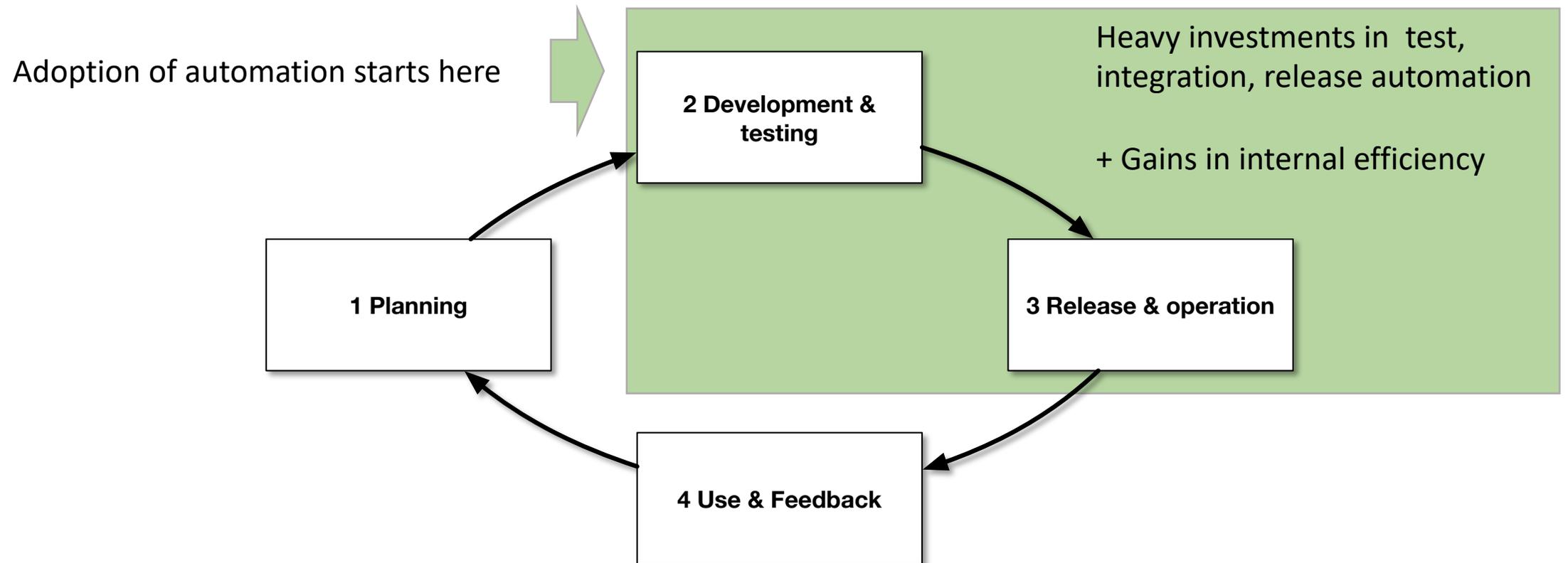


Case Studies

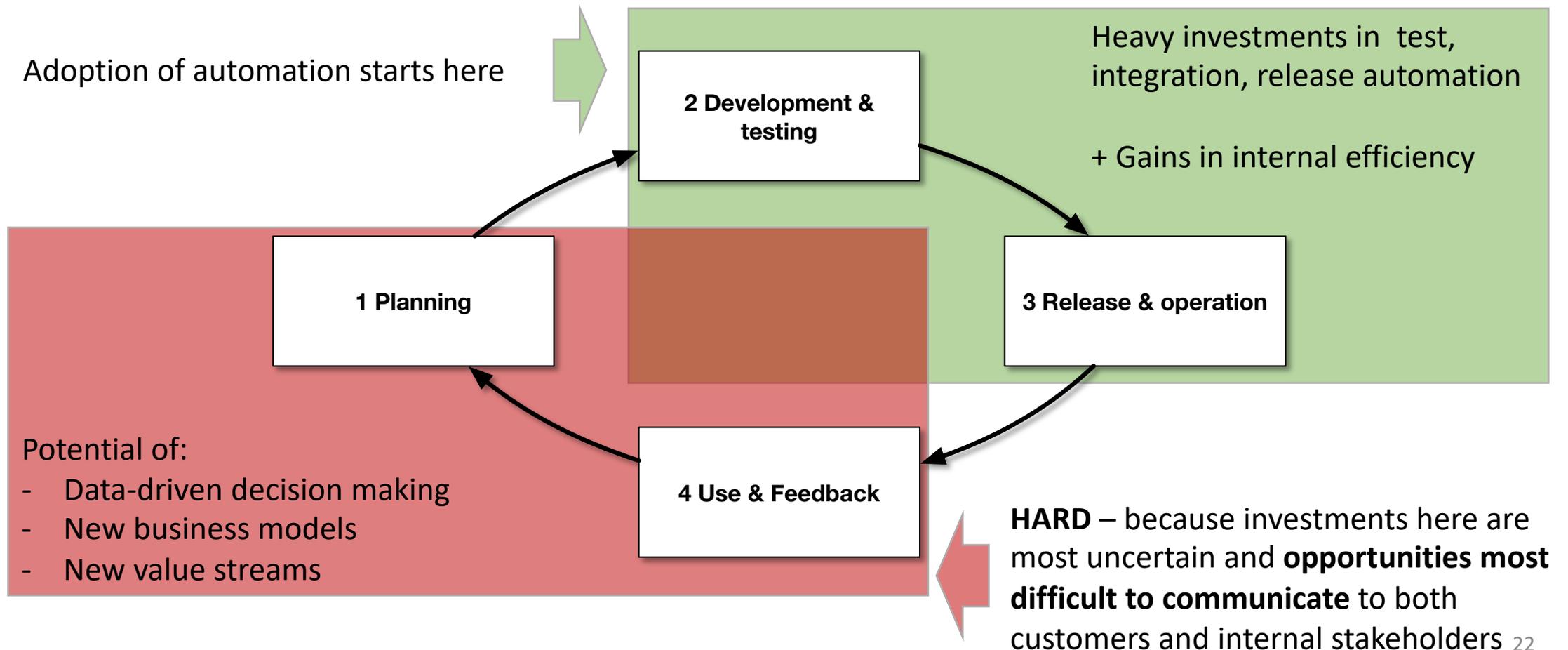
Use the conceptual model as blueprint for adopting CI/CD



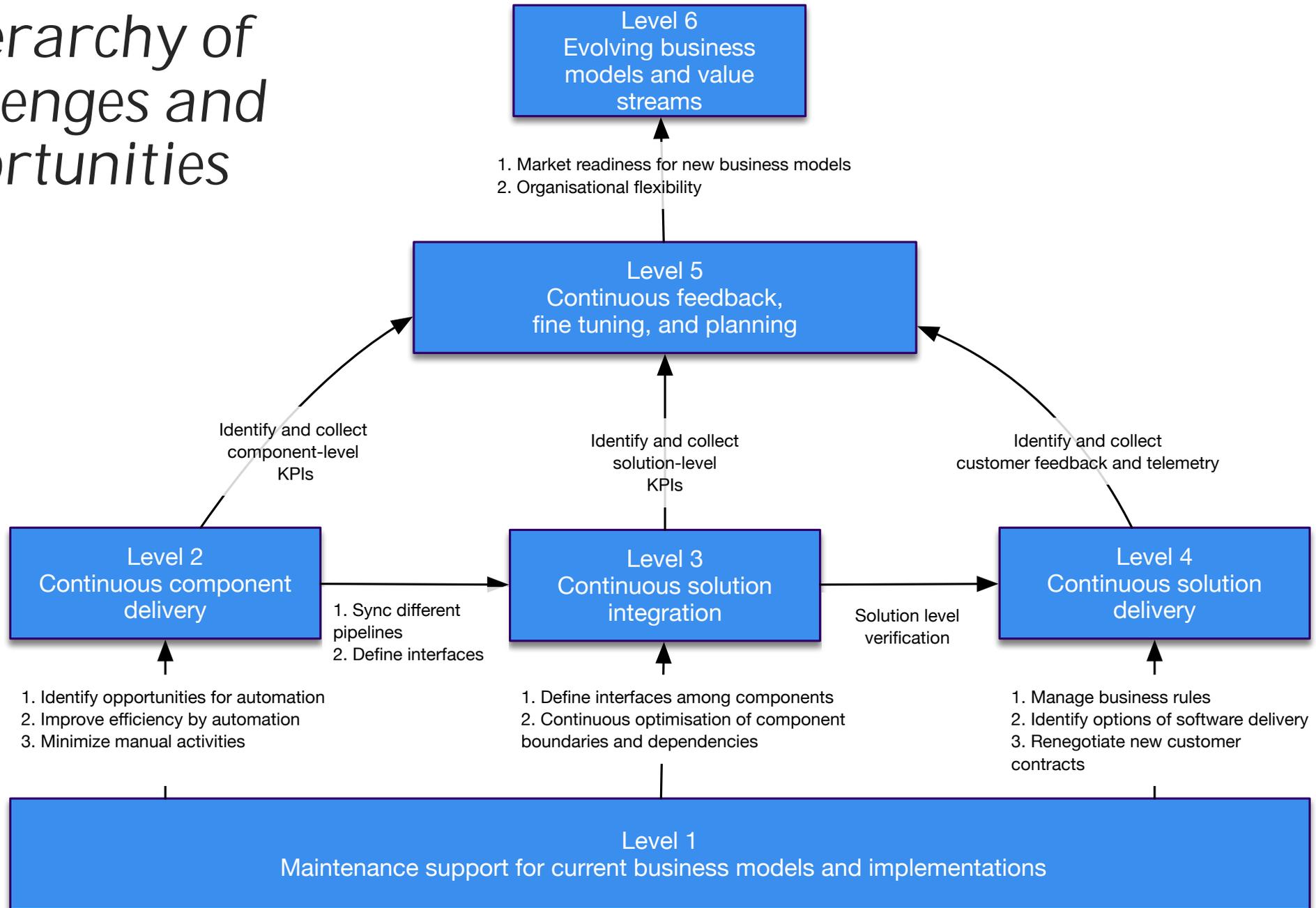
Case Studies



Case Studies



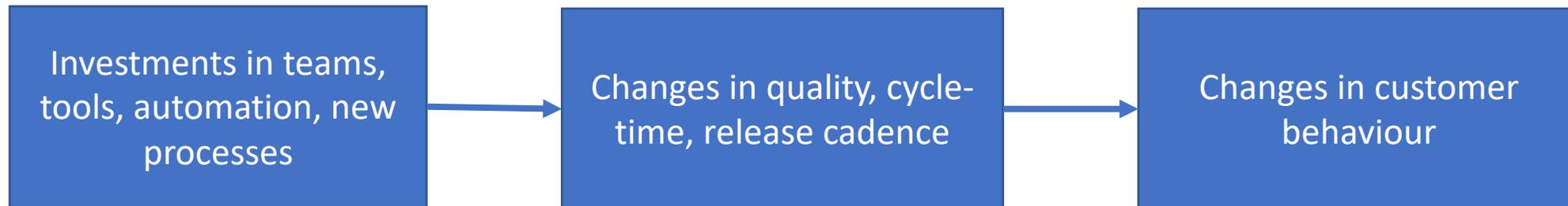
A hierarchy of challenges and opportunities



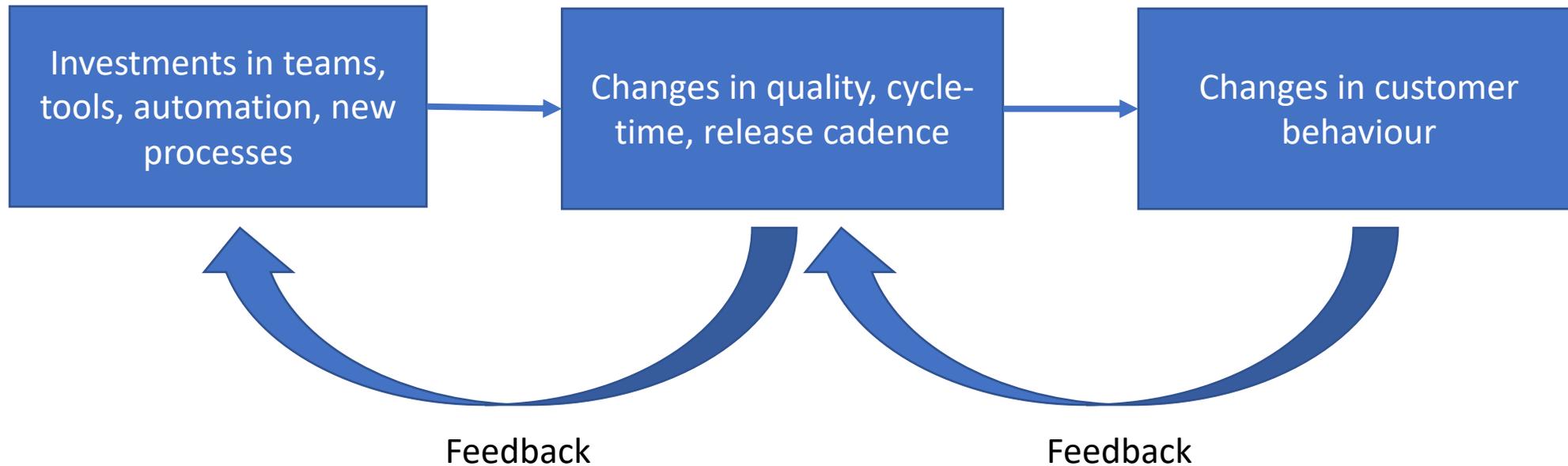
A man in a dark racing shirt with logos like 'VOLVO OCEAN RACE' and 'SAILOR RACING' is operating a sailboat on the ocean. He is looking towards the camera. In the background, another person is visible at the helm. The boat has various equipment, including a Harken winch and a Volvo Penta outboard motor. The sky is overcast.

Calibrating the organization

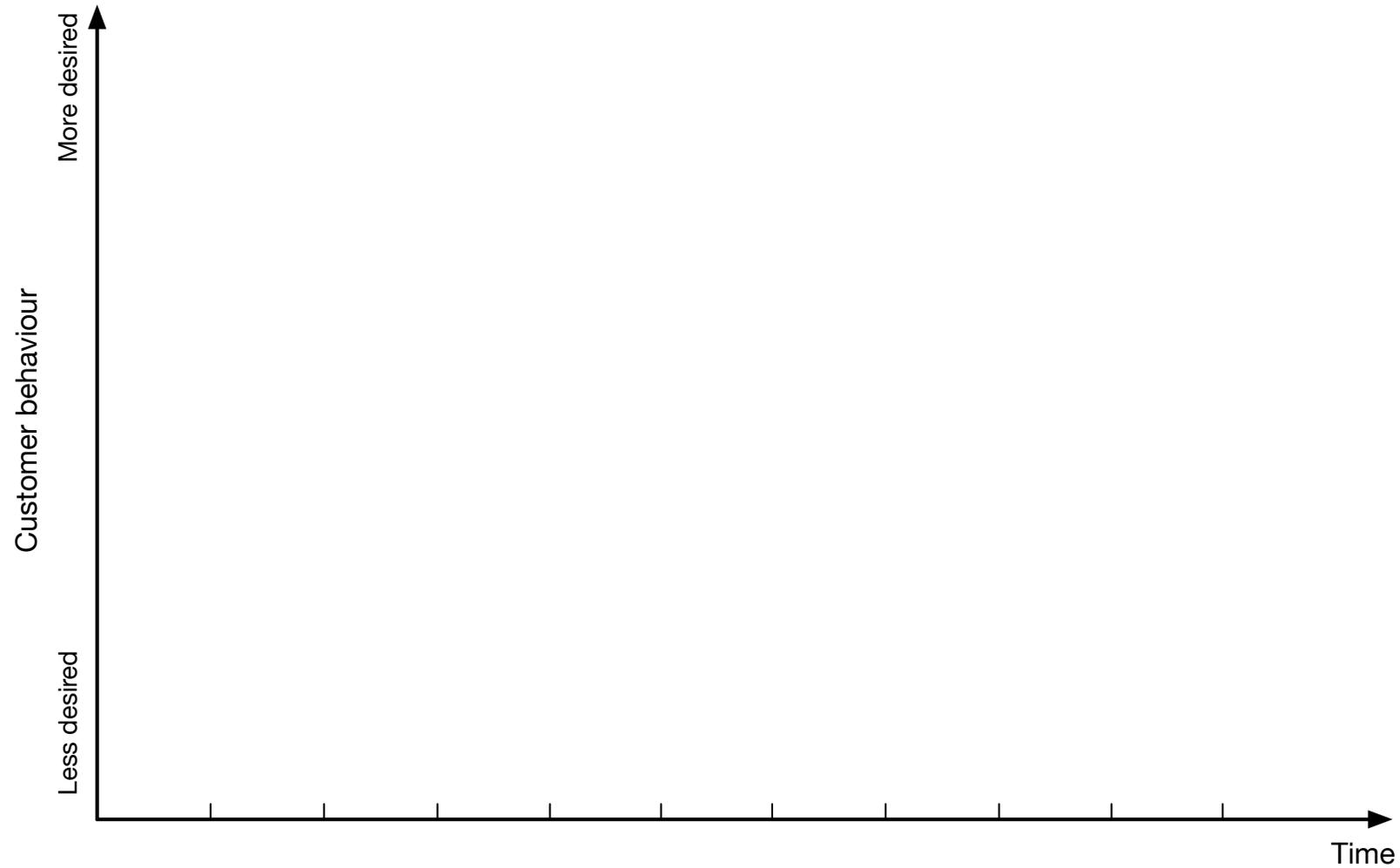
Calibrating the organization



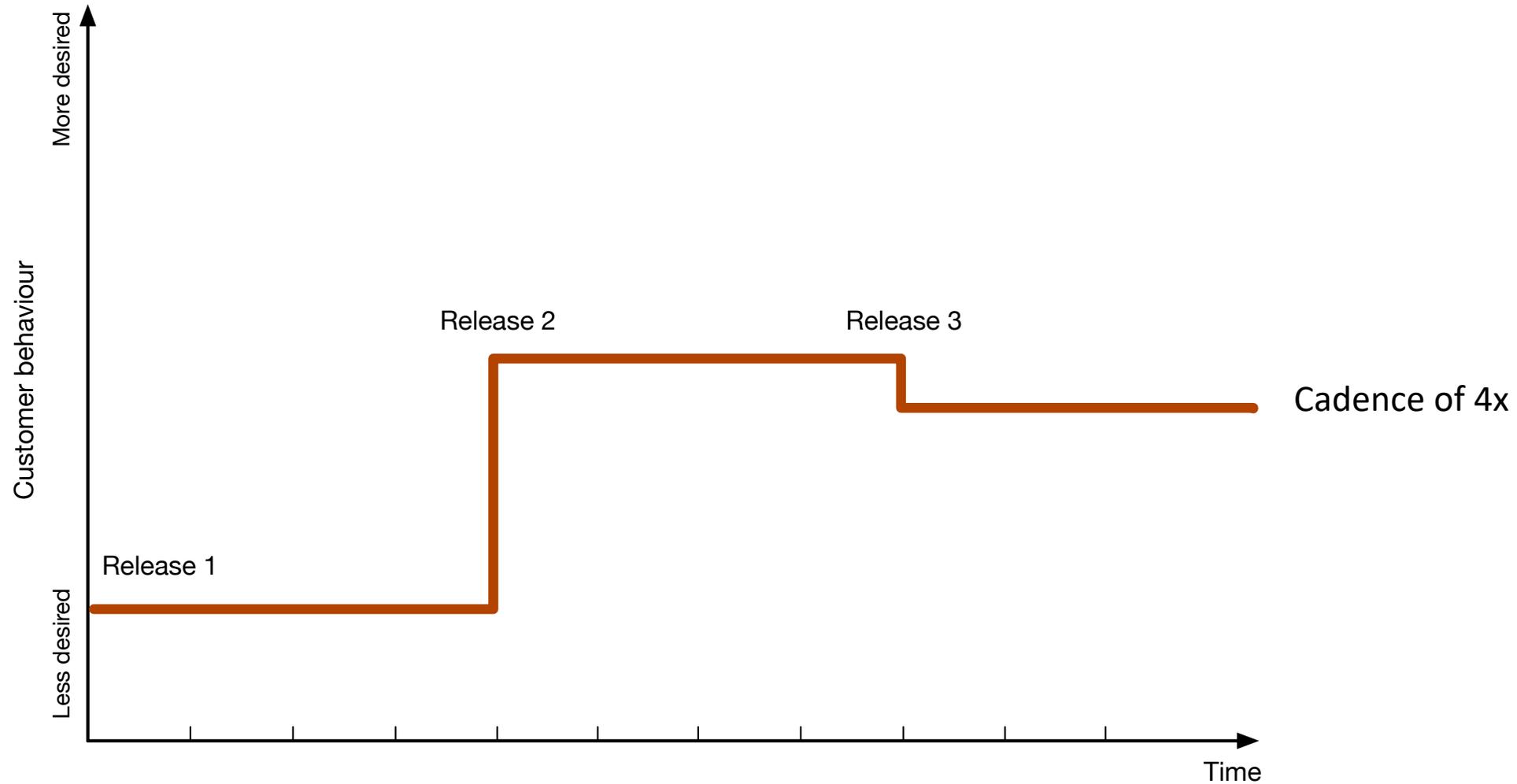
Calibrating the organization



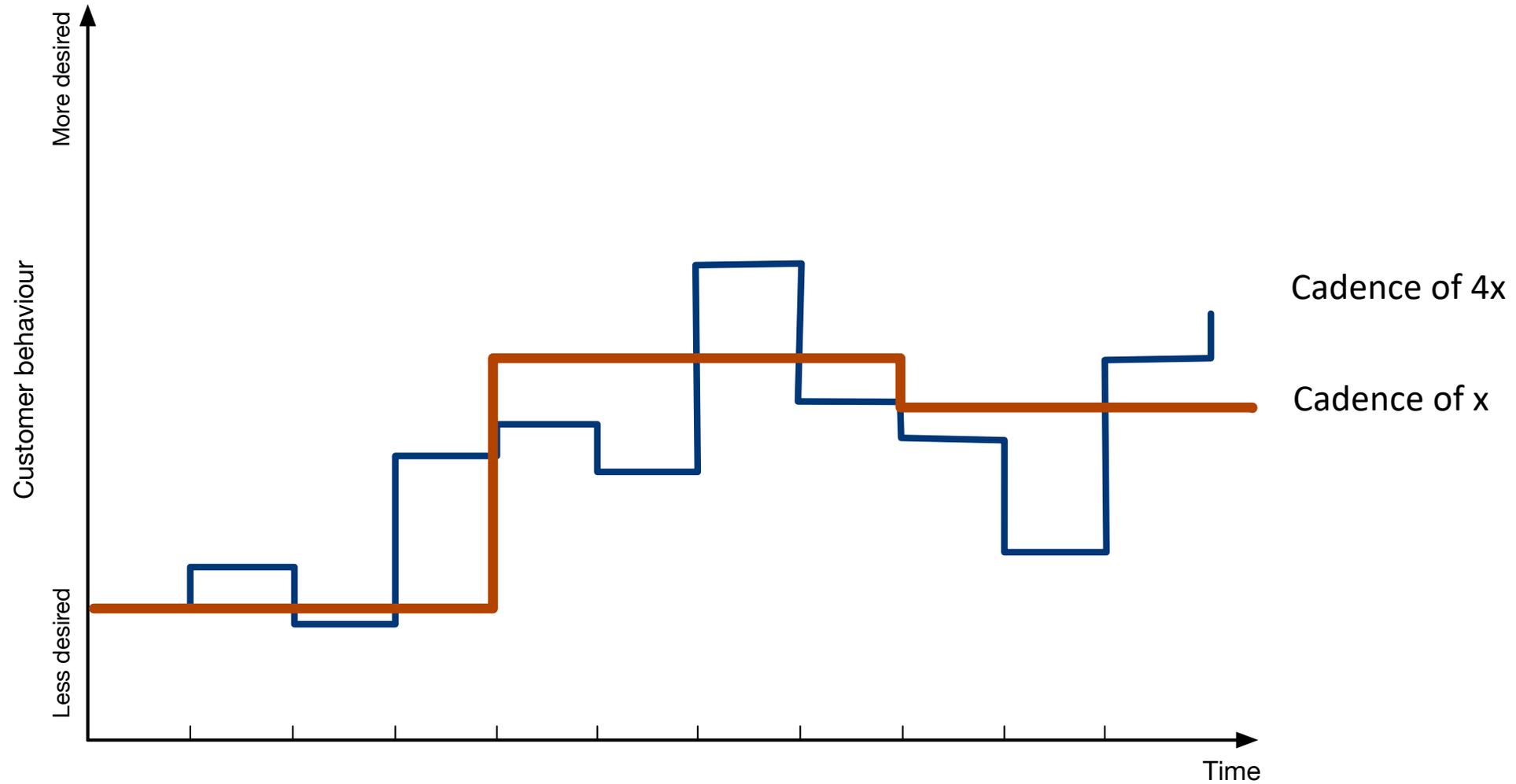
Calibrating the release cadence



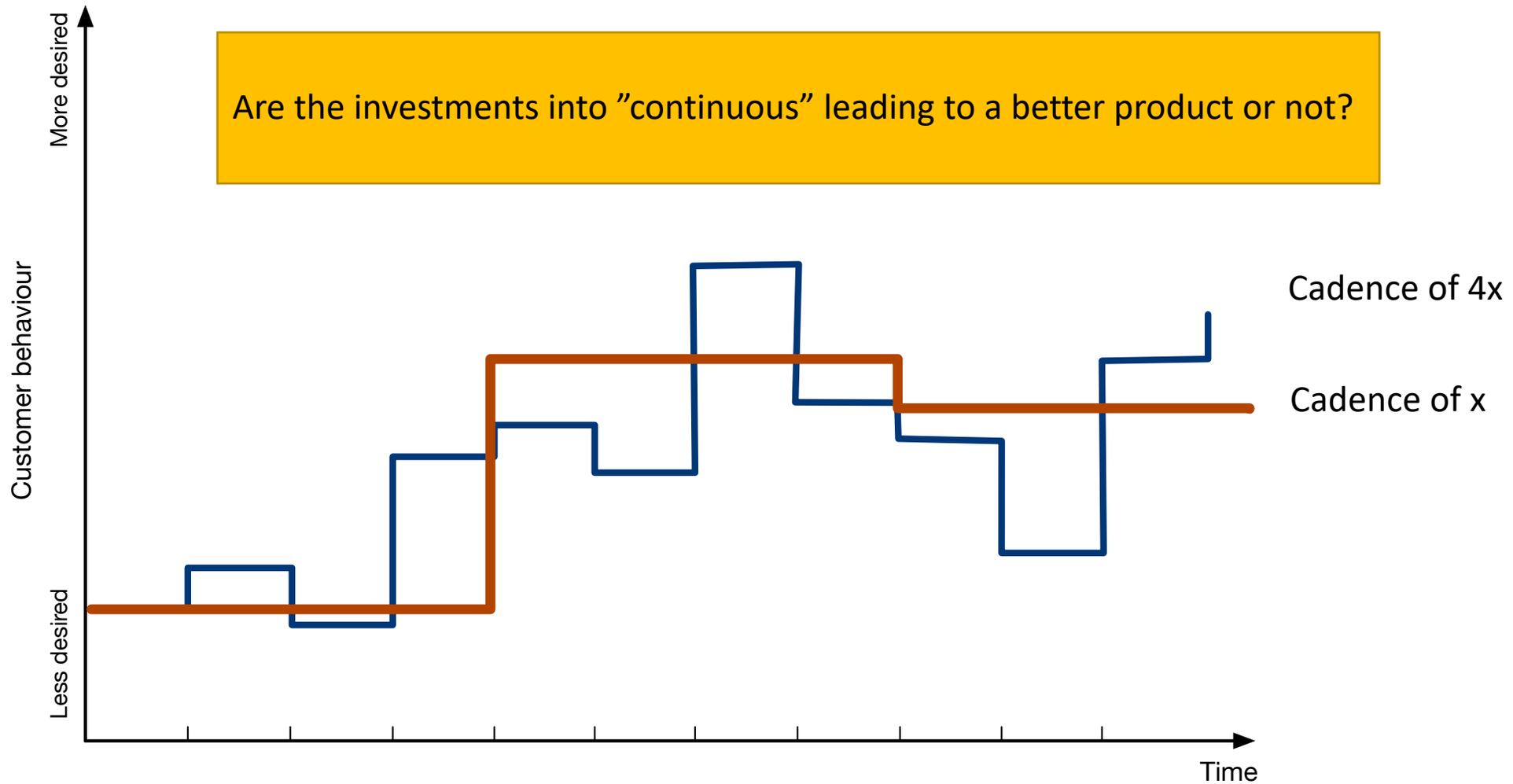
Calibrating the release cadence



Calibrating the release cadence



Calibrating the release cadence



Interested?

1. Inventory and evaluate your current situation. What are the next steps towards continuous software delivery pipeline?
 - A workshop with your software delivery stakeholders + a follow up meeting to discuss the results
2. An assessment of your software delivery metrics. What is the alignment between the metrics and decisions to be supported?
 - Two workshops + a follow up meeting

We are open to collaboration!

*A picture to
remember*



gettyimages®
coolbiere photograph



Get in touch!

Eriks Klotins, PhD

Post-doctoral researcher

@ Blekinge Institute of Technology

eriks.klotins@bth.se

<https://www.linkedin.com/in/eriksklotins/>

BTH **SERL Sweden**
LEADING SOFTWARE ENGINEERING

S.E.R.T
reThink•reThought