

# *Costs and Benefits of Continuous Everything*

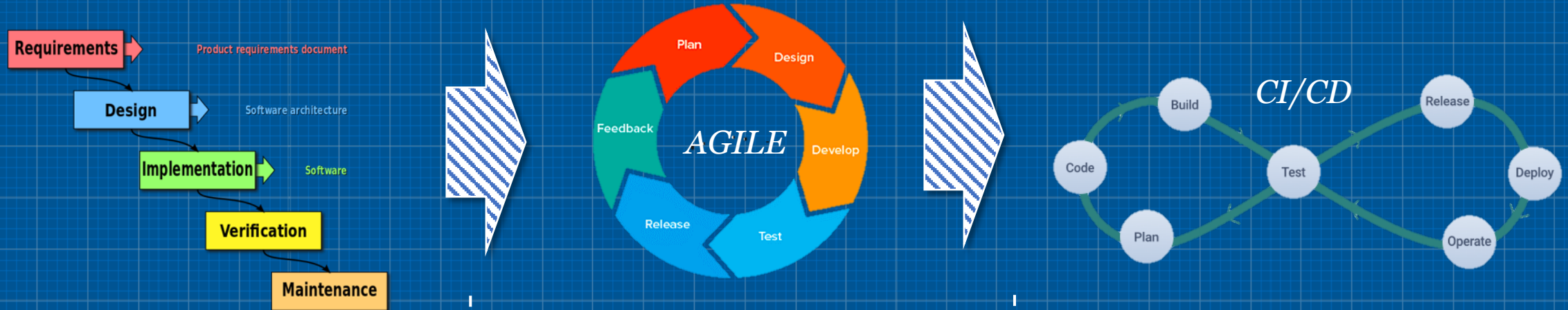
Eriks Klotins



**BTH** SERL Sweden  
LEADING SOFTWARE ENGINEERING



# Movement Towards Continuous SE



- Release time in months/years
- All project value is delivered at the end
- It may take years to identify and correct a problem
- Relies on upfront process & planning

- Release every 2-4 weeks
- Value is delivered in chunks throughout the project
- It may take a few weeks to fix a problem
- Relies on flexible collaboration

- Release immediately
- Value is delivered in a continuous flow
- It may take minutes to deliver a feature and receive feedback
- Relies on automation, data, and analytics

# Related work

## Continuous Delivery Huge Benefits, but Challenges Too

Lianping Chen, Paddy Power

### The Context

Paddy Power is a company, with approximately €6 million in revenue and 100 employees. It operates in regulated markets such as shops, phones, and online services. The company is facing an increasingly large market for custom software applications including...

## Continuous Delivery? Easy! Just Change Everything (well, maybe it is not that easy)

Steve Neely and Steve Stolt  
Rally Software  
3333 Walnut Street  
Boulder, Colorado 80301  
Email: {sneely, stolt}@rallydev.com

**Abstract**—Rally Software transitioned from shipping code every eight-weeks, with time-boxed Scrum sprints, to a model of continuous delivery with Kanban. The team encountered complex challenges with their build systems, automated test suites, customer enablement, and internal communication. But there was light at the end of the tunnel — greater control and flexibility over feature releases, incremental delivery of value,

goes straight to production. The frequency is not our deciding factor. It is the ability to deploy at will.

Covertly, we know that the engineering discipline required to support such a model must be strong. Our systems must be finely tuned and comprehensively governed by automation, with thorough monitoring and testing frameworks.



ELSEVIER

The Journal of Systems and Software 128 (2017) 72–86

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: [www.elsevier.com/locate/jss](http://www.elsevier.com/locate/jss)



Continuous Delivery: Overcoming adoption challenges<sup>☆</sup>



Lianping Chen

Lianping Chen Limited, Dublin, Ireland

ARTICLE INFO

ABSTRACT

Article history:  
Received 29 August 2016

Continuous Delivery (CD) is a relatively new software development approach. Companies that have adopted CD have reported significant benefits. However, the adoption of CD is not without challenges. This paper discusses the challenges of adopting CD and provides a framework for overcoming these challenges.



ELSEVIER

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: [www.elsevier.com/locate/jss](http://www.elsevier.com/locate/jss)



Continuous software engineering: A roadmap and agenda



Brian Fitzgerald, Klaas-Jan Stol<sup>\*</sup>

Lero—the Irish Software Research Centre, University of Limerick, Ireland

ARTICLE INFO

ABSTRACT

Article history:  
Received 11 November 2014

Throughout its short history, software development has been characterized by harmful disconnects between important activities such as planning, development and implementation. The problem is further exacerbated by the increasing complexity of software systems and the need for more frequent releases. This paper discusses the challenges of software development and provides a roadmap and agenda for continuous software engineering.

Jan Bosch Editor

# Continuous Software Engineering

Springer

THE SCIENCE OF LEAN SOFTWARE AND DEVOPS  
**ACCELERATE**  
Building and Scaling High Performing  
Technology Organizations

Nicole Forsgren, PhD  
Jez Humble, and Gene Kim

with forewords by Martin Fowler and Courtney Kissler  
and a case study contributed by Steve Bell and Karen Whitley Bell

# *The Opportunity*

1. By a magnitude shorter feedback loop with customers
2. More capacity for discovering new features with experimentation
3. Customer feedback & telemetry to support data-driven decisions
4. New business models and value streams



*Exponential advantage in the market*

# *The Challenge*

1. What is the investment?
2. How can we incrementally benefit from CI/CD?
3. What are all the benefits we can aim for?
4. How to estimate the impact from organizational change?
5. How to get it right from the cost-benefit perspective?

*Substantial and uncertain costs of implementation and then maintenance is a barrier in adopting CI/CD*

# *Our Aims & Objectives*

## *Aim:*

A blueprint for CI/CD cost-benefit calculation & modelling

## *Objectives of the blueprint:*

1. Provide cost-benefit analysis of CI/CD activities
2. Support fine-tuning and synchronization of different activities & pipelines
3. Model and maximize the benefits for a specific case

# *The World Map*

*Engineering*

*Product*

*Operations*

*End-user / customer*

# *The World Map*

*Engineering*



*1 Continuous  
product planning*

*Product*



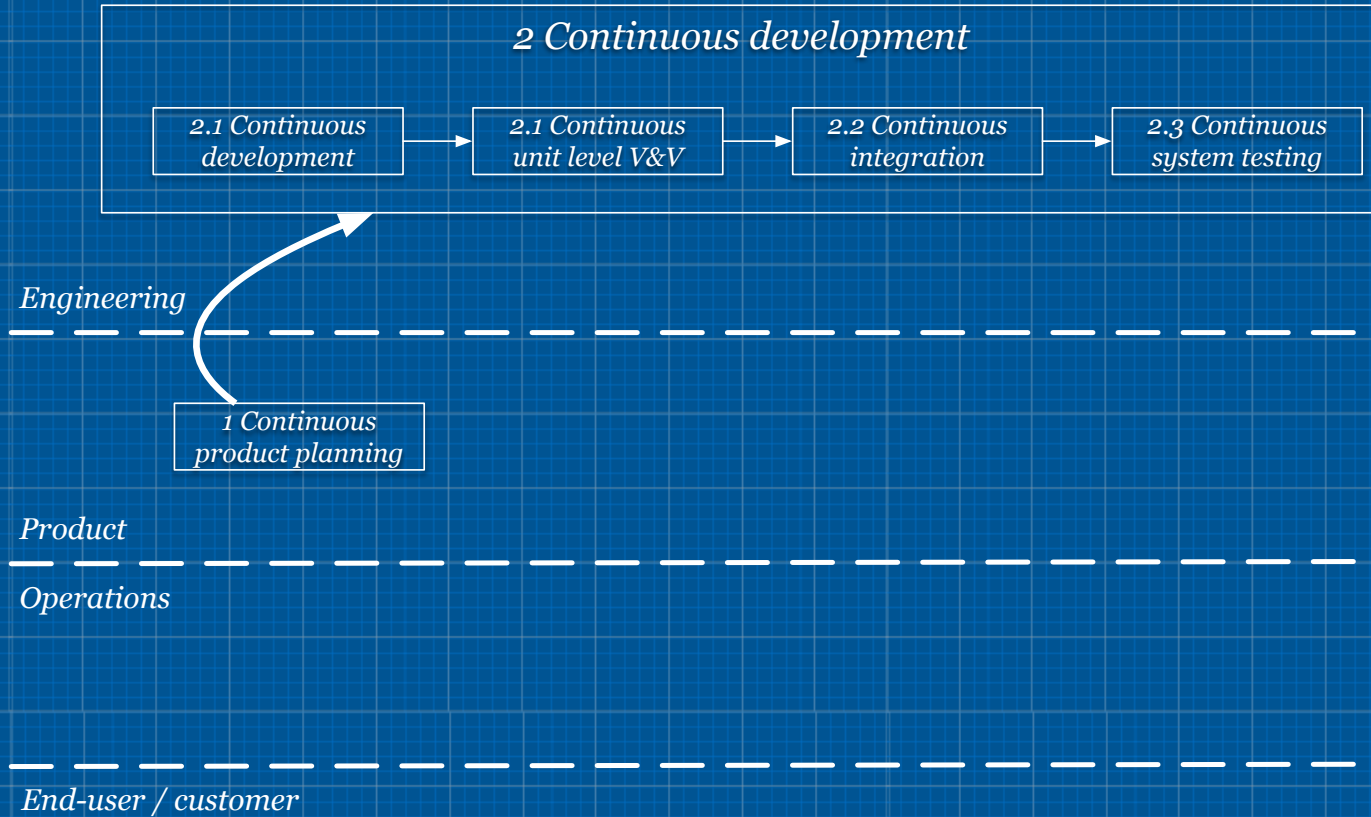
*Operations*



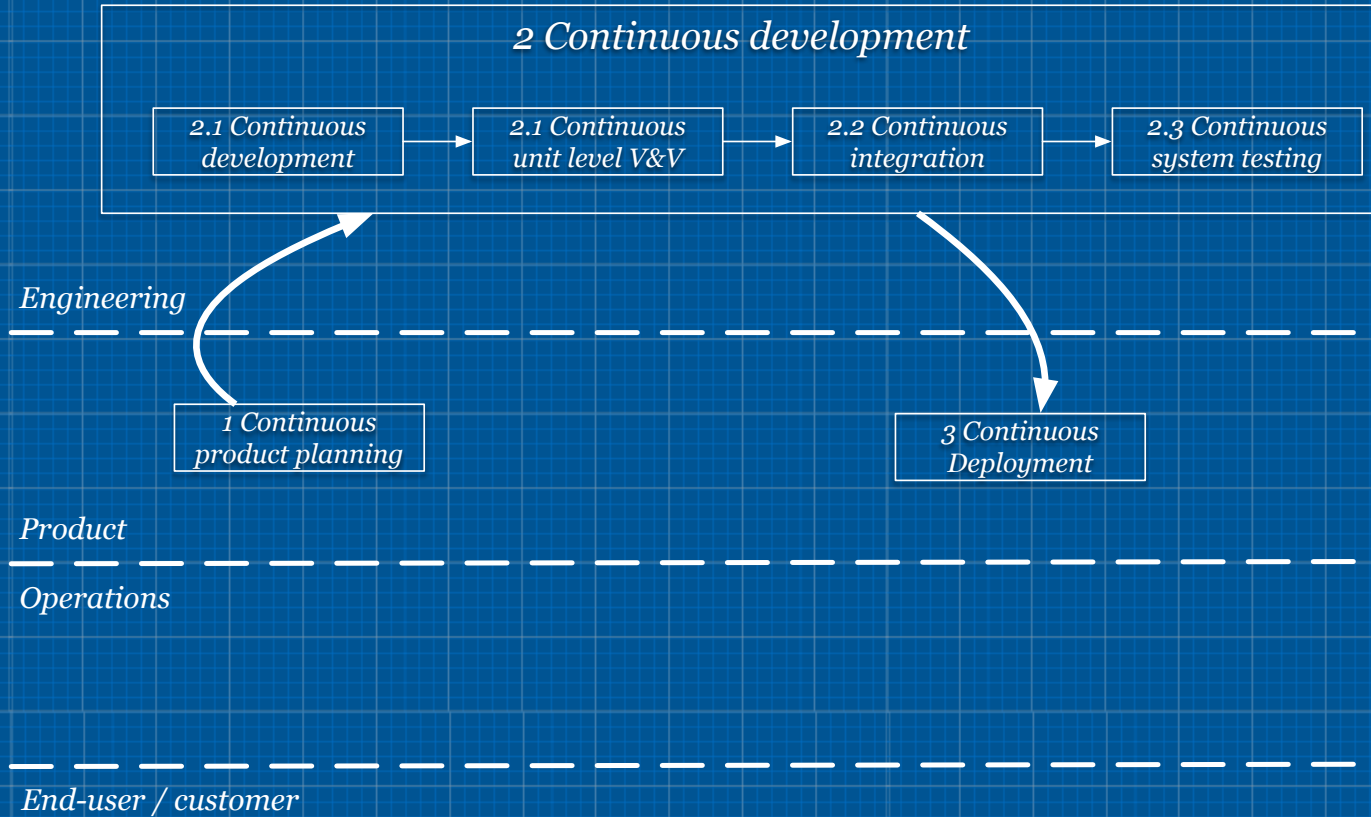
*End-user / customer*



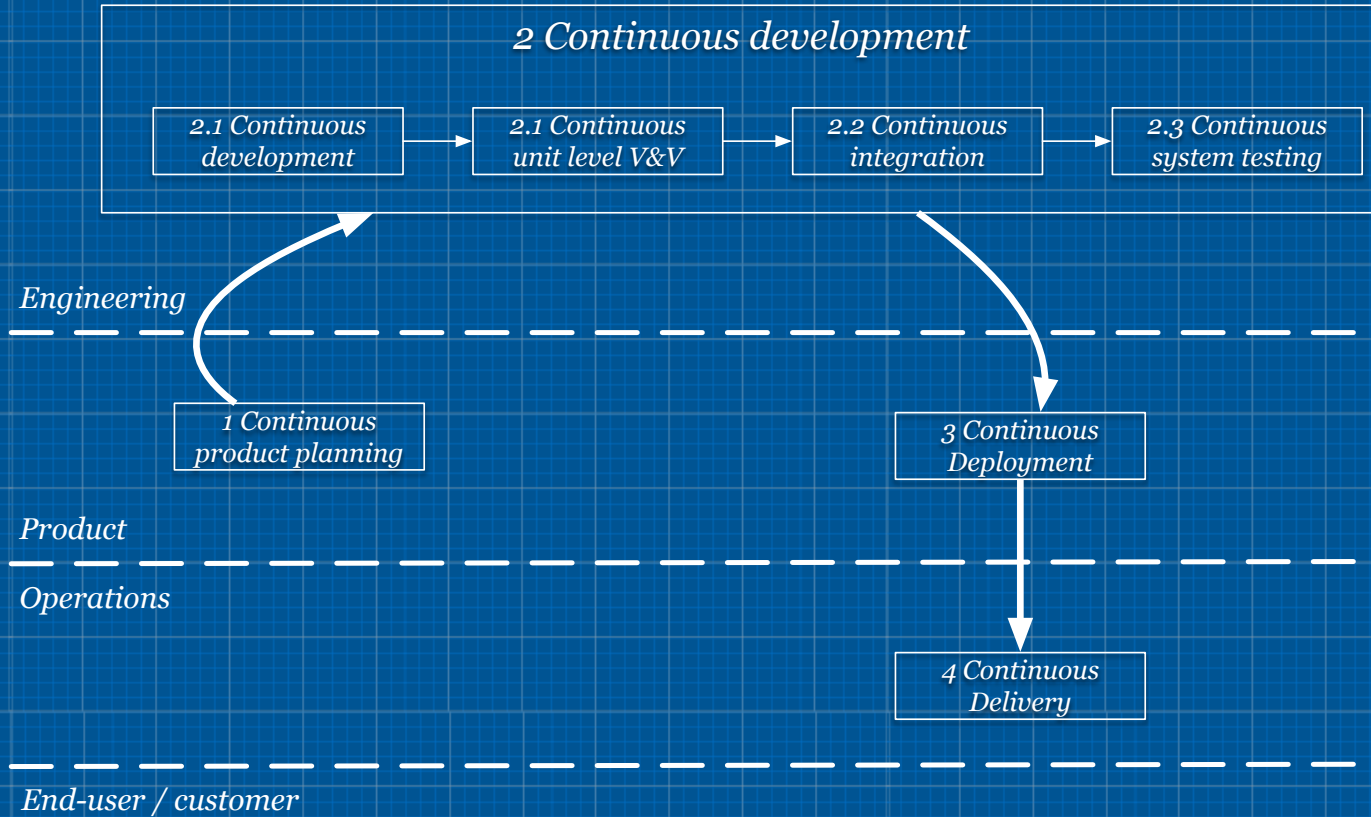
# The World Map



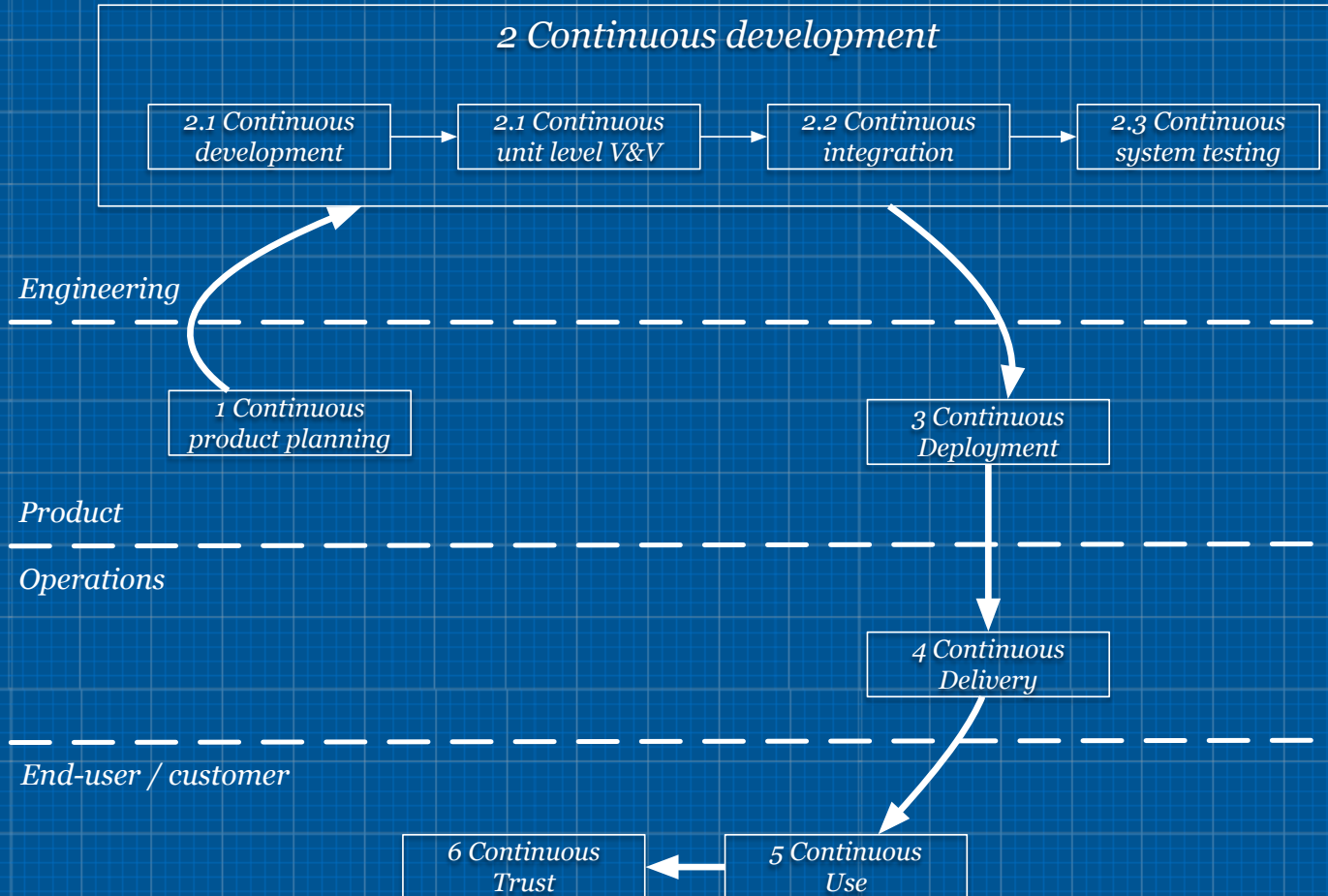
# The World Map



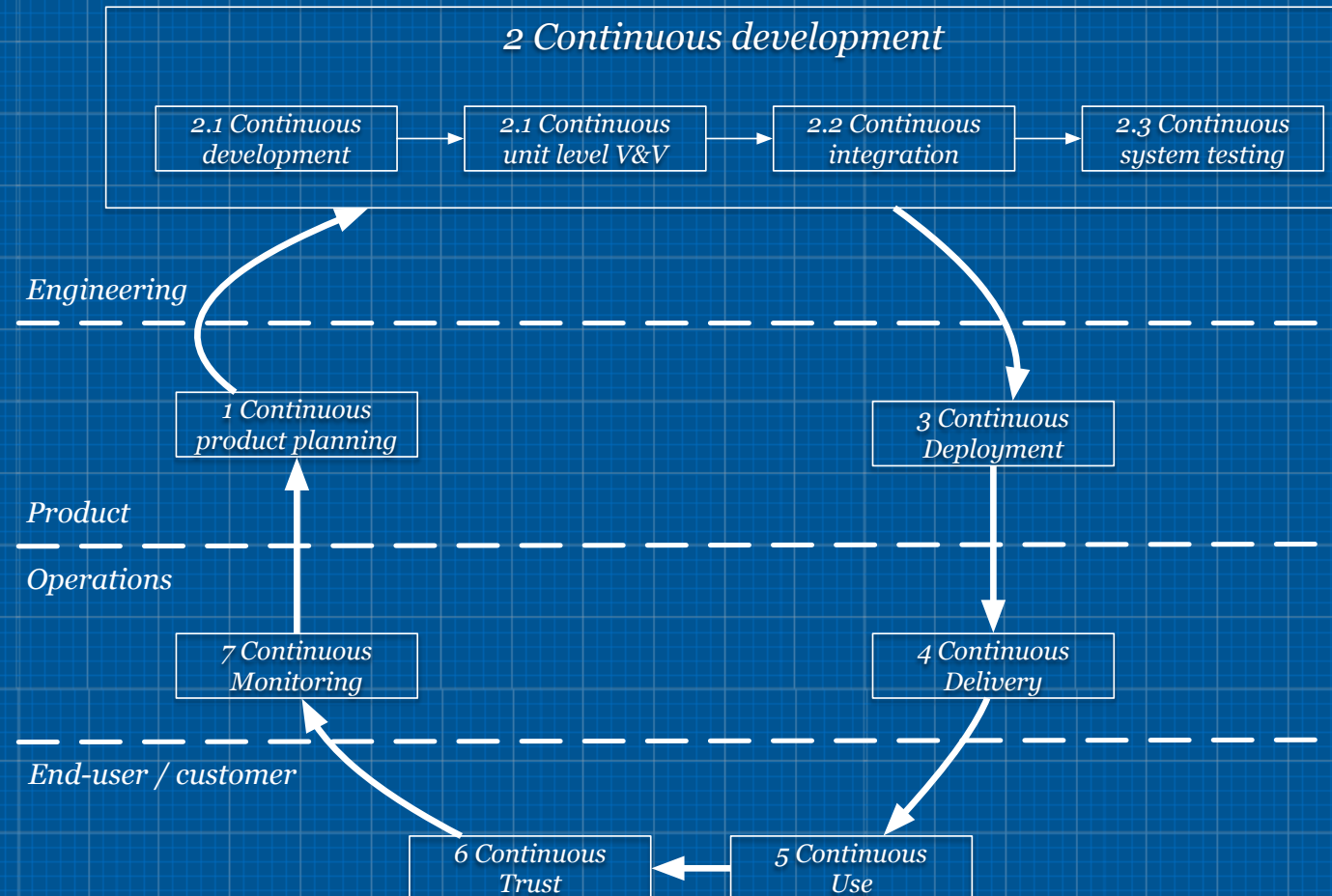
# The World Map



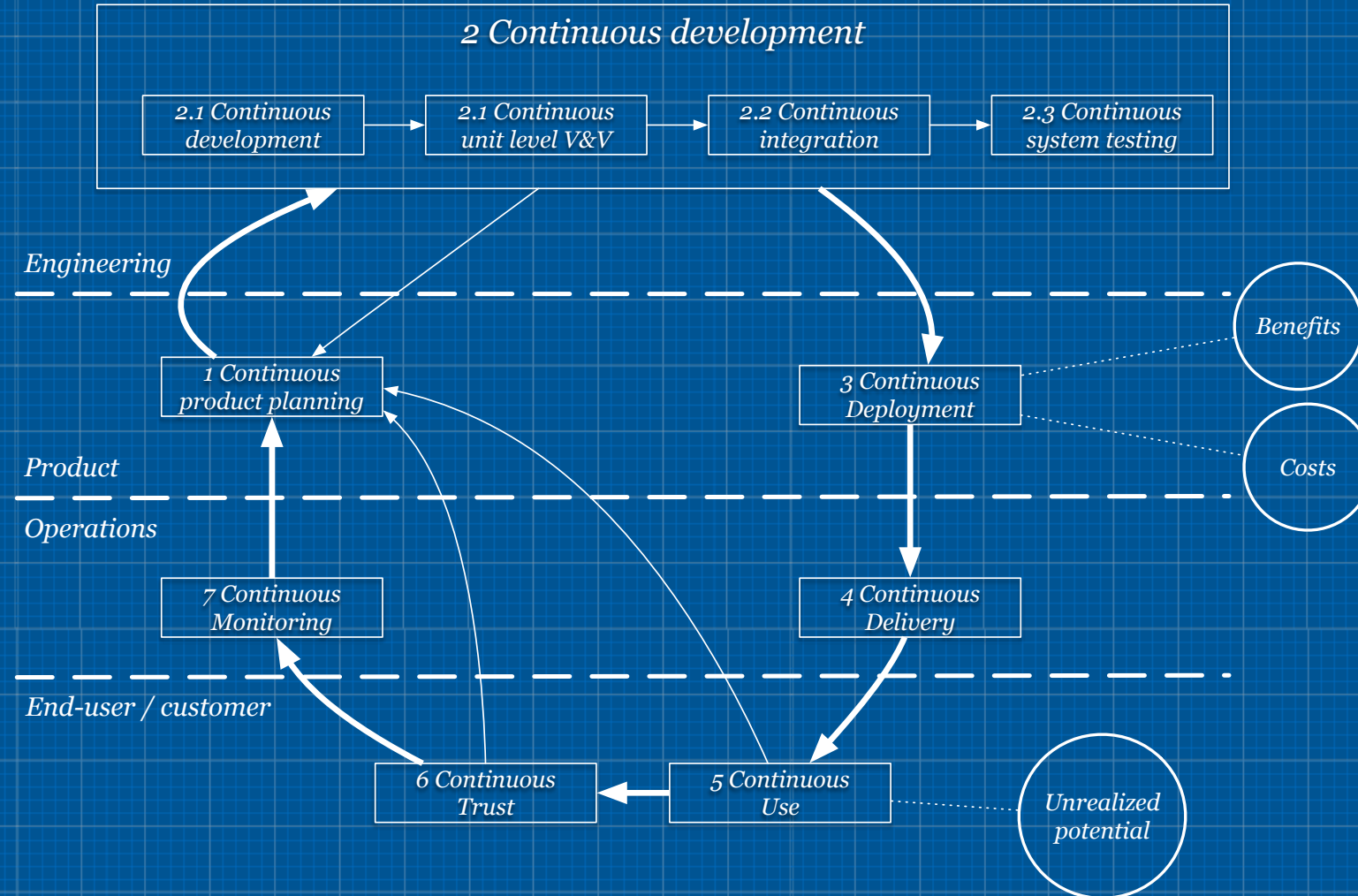
# The World Map



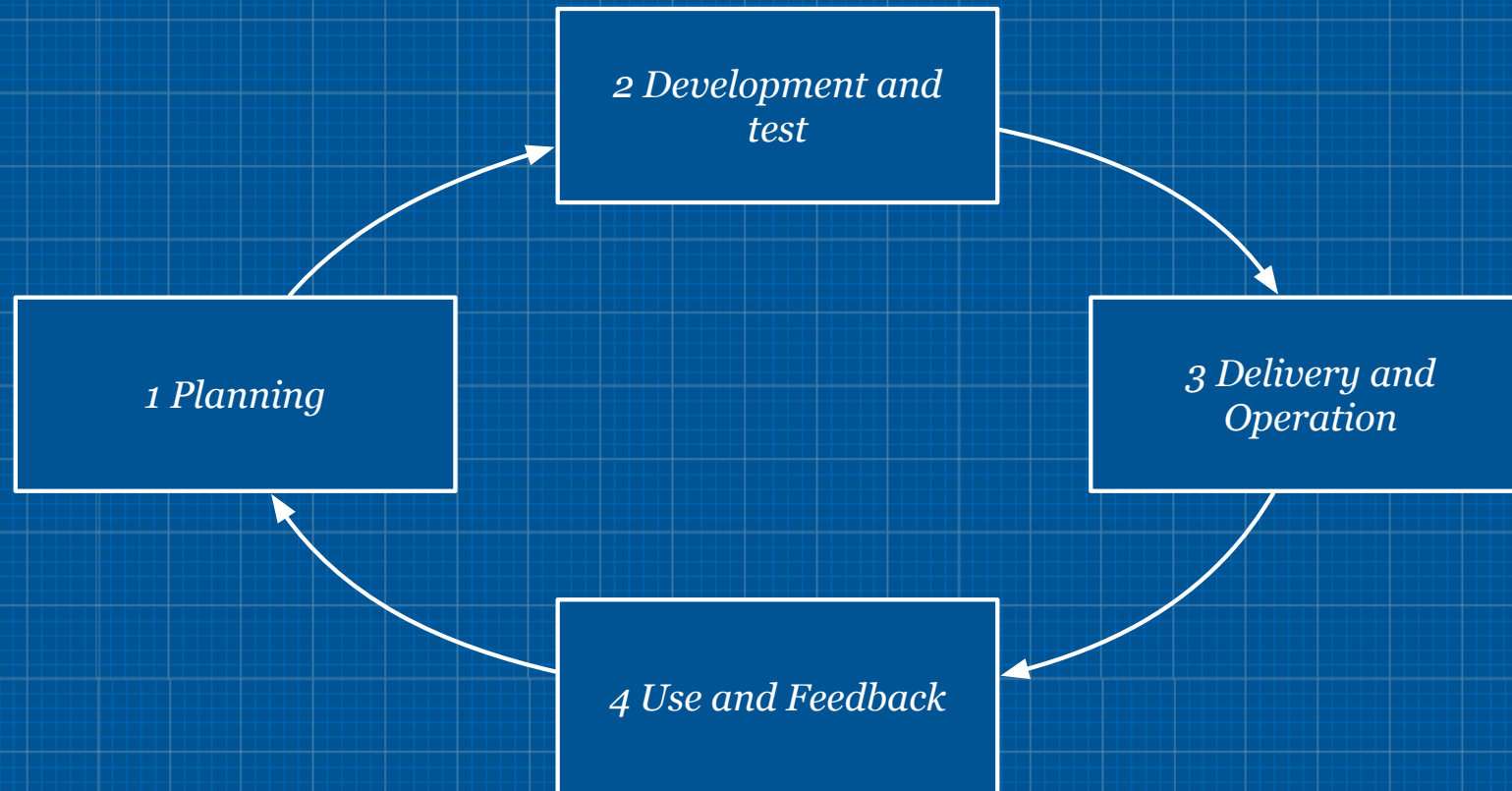
# The World Map



# The World Map



# *The Simplified Model*



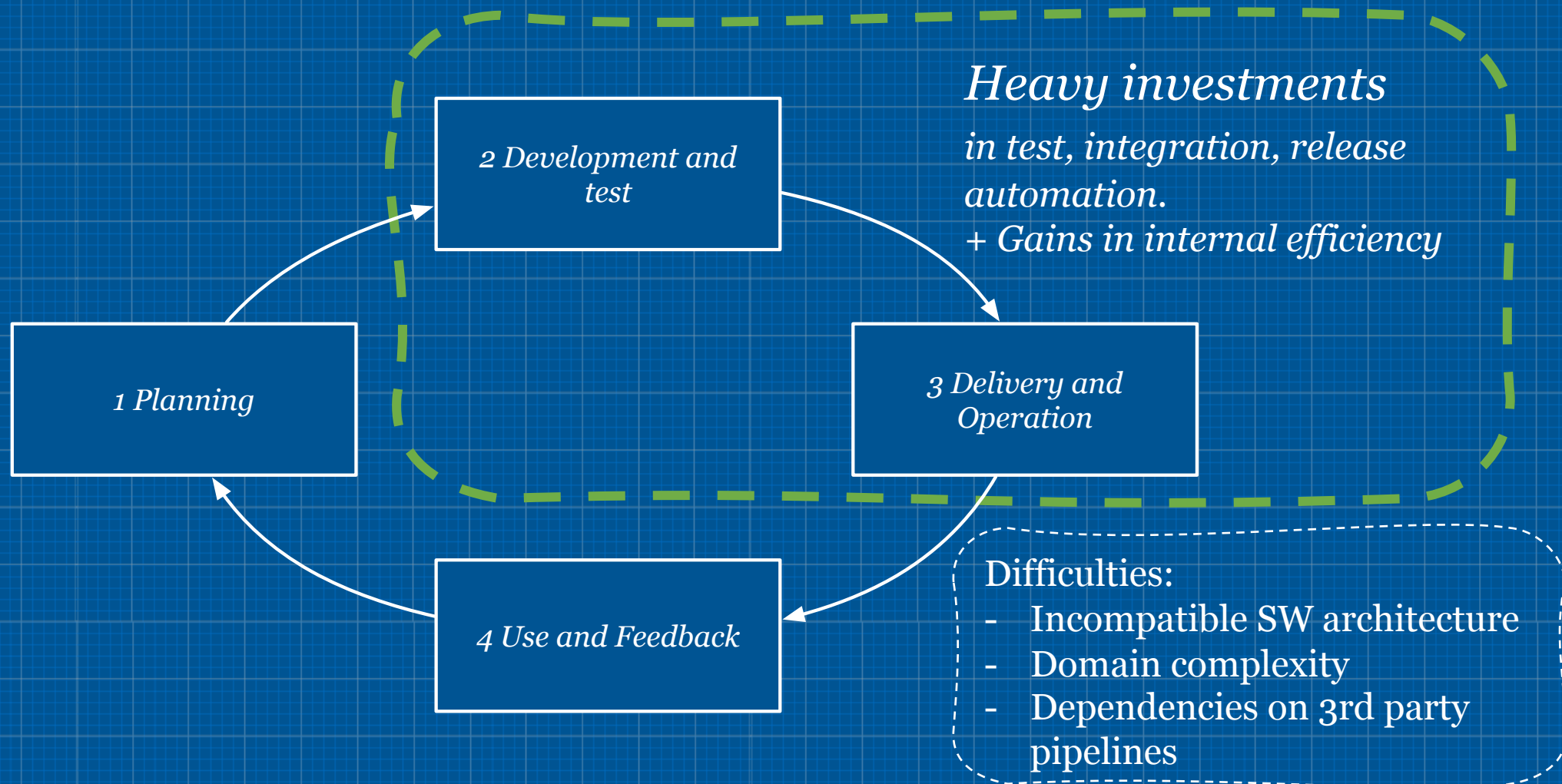
Present in any organization, continuous or not

# *Ongoing research work*

1. A catalog of cost and benefit items throughout the pipeline
  - Knowledge of what can be achieved helps to achieve it
  - Highlights unrealized potential in the pipeline
2. Estimation methods for costs and benefits
  - Helps to pinpoint high benefit/low cost items
  - Enables cost-benefit calculation for modelling and fine-tuning



# A Typical Case So Far



# A Typical Case So Far

HARD – because costs here are most uncertain and opportunities most difficult to communicate. Also, most benefits lay here

1 Planning

2 Development and test

3 Delivery and Operation

4 Use and Feedback

*Heavy investments in test, integration, release automation.  
+ Gains in internal efficiency*

*Unrealized potential from:*

- *Data-driven decision making*
- *«Getting out of the box»*
  - *New business models*
  - *New value streams*

# The Cost Benefit Modelling

## Metrics:

### Benefits: *(all that is desired)*

- Internal value (Efficiency etc)
- External value (Customer value)
- New opportunities and more..

### Costs: *(all that is to be minimized)*

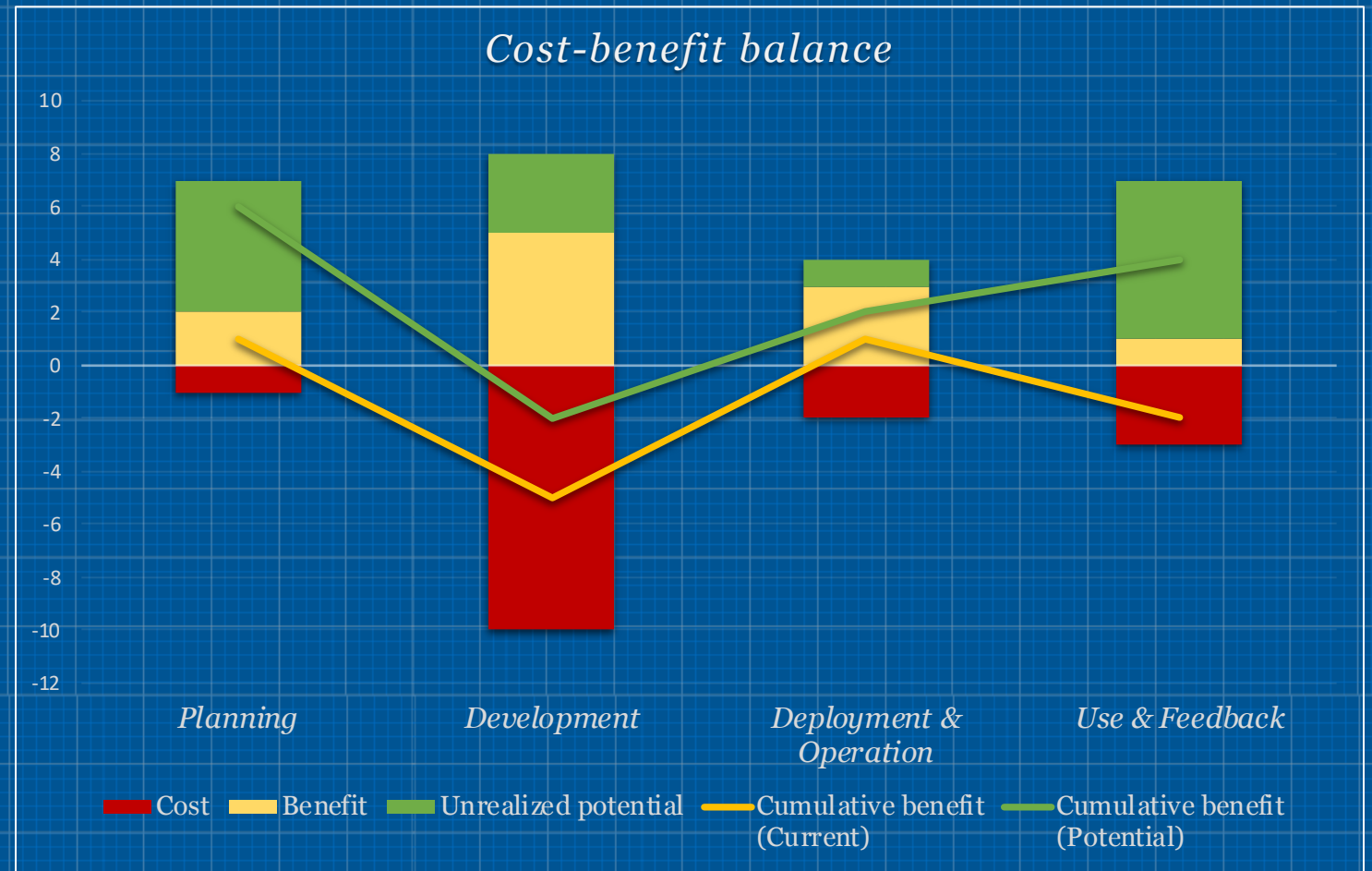
- Cost/\$\$\$
- Time
- Waste
- Defects and more..

### Control:

- Cycle-time, Lead time, time to ..

### Scope:

- One time
- Per-cycle
- Persistent



# Use Cases

1. *Visualize the specific costs and benefits of implementation and maintenance of the pipeline*
  - *Input for decision support*
  - *ROI/TTROI analysis*

4. *Understand the downsides of continuous software delivery*
  - *Avoid the bandwagon*
  - *Decision support*
  - *Risk assessment*

Known costs	Realised benefits
Hidden costs	Unrealized benefits

2. *Measure current implementation of the pipeline*
  - *Fine-tuning to maximize benefits*
  - *Input for decision making*

3. *Highlight untapped potential from the pipeline*
  - *A guide the further adoption of continuous practices*
  - *Maximize the overall benefit*

# *Collaboration Model*

## Low hanging fruit:

Inventory and analysis of your “continuous” aspirations, practices, costs, benefits, challenges

- Input for our research
- External assessment of your current status

Typically takes 2-3 workshops

# *Questions?*

Eriks Klotins

[eriks.klotins@bth.se](mailto:eriks.klotins@bth.se)