The Big Picture of Continuous Everything

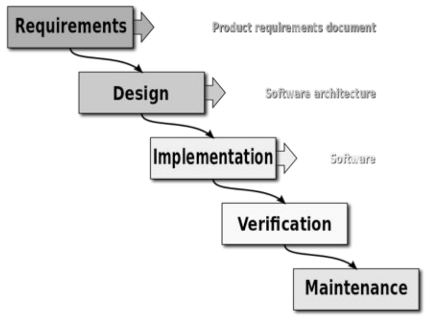
Eriks Klotins

BTH SERL Sweden LEADING SOFTWARE ENGINEERING

S.E.R.T reThink▪reThought

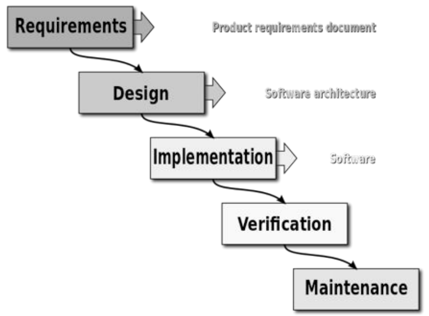
Everyone Wants to Get Better At Delivering Software

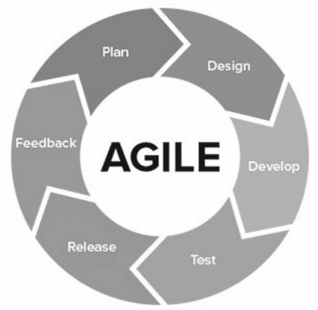
From Plan-driven, to Agile, to Continuous



- Release time in years
- All project value (and risk) is delivered at the end
- It may take years to identify and fix a problem
- Relies on upfront process & planning

From Plan-driven, to Agile, to Continuous

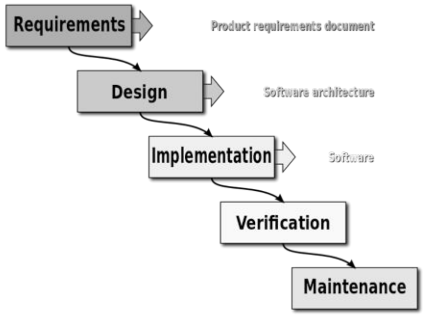


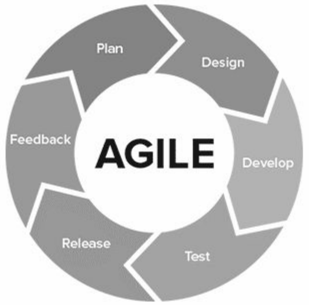


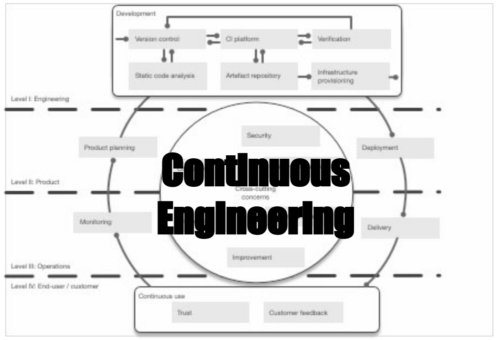
- Release time in years
- All project value (and risk) is delivered at the end
- It may take years to identify and fix a problem
- Relies on upfront process & planning

- 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 and a customer in the room

From Plan-driven, to Agile, to Continuous







- Release time in years
- All project value (and risk) is delivered at the end
- It may take years to identify and fix 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 and a customer in the room

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

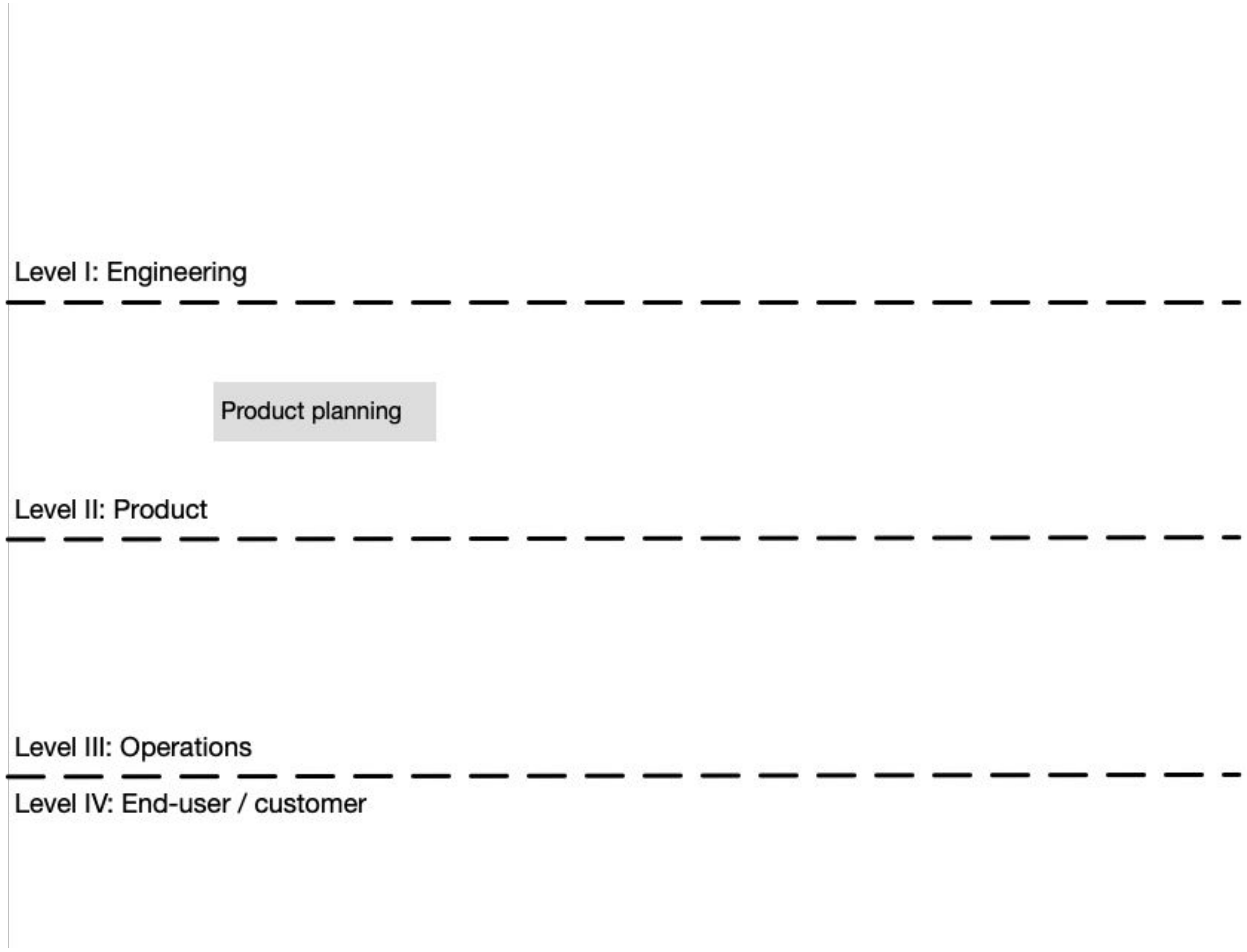
State-of-the-Art
Continuous Software Engineering

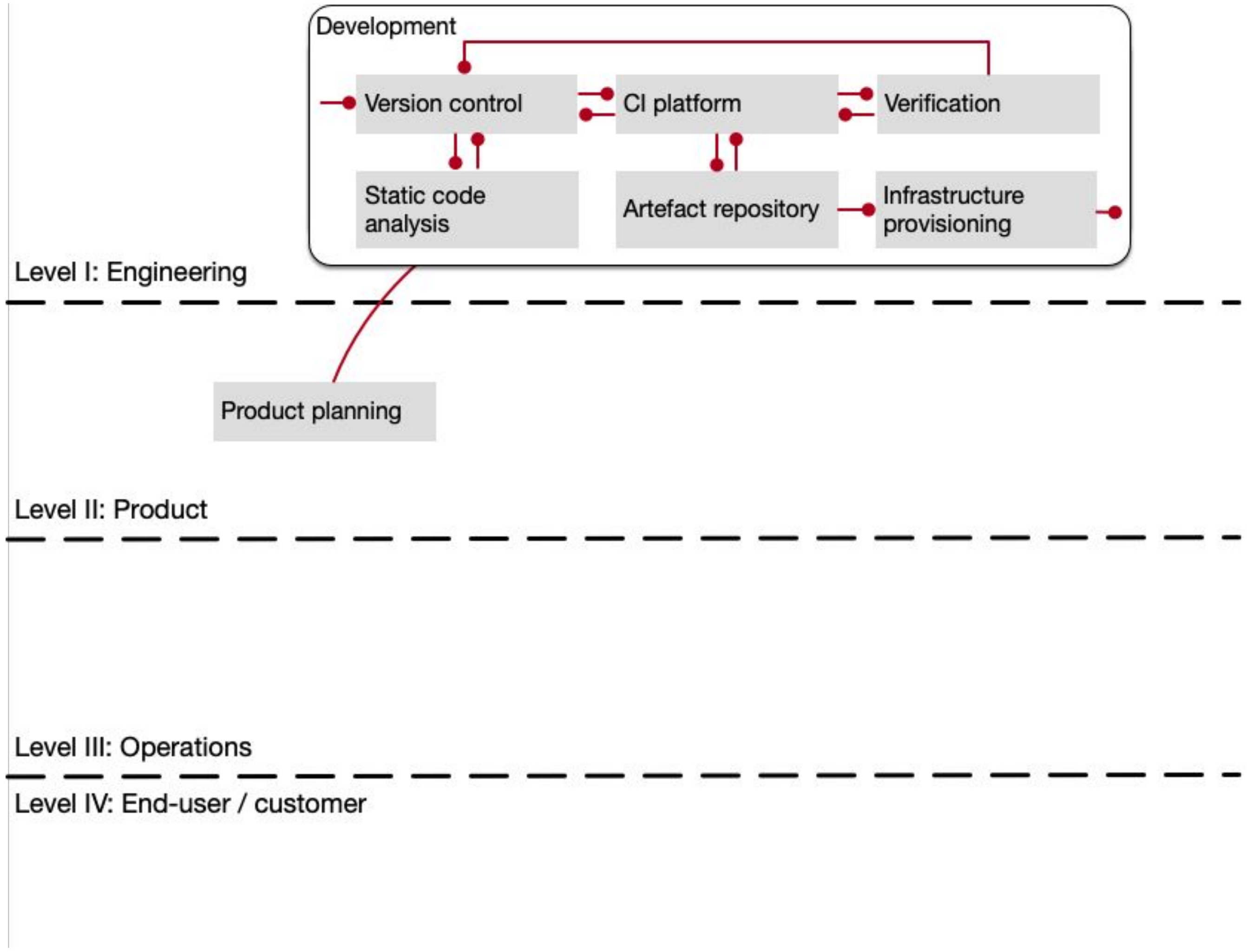
Level I: Engineering

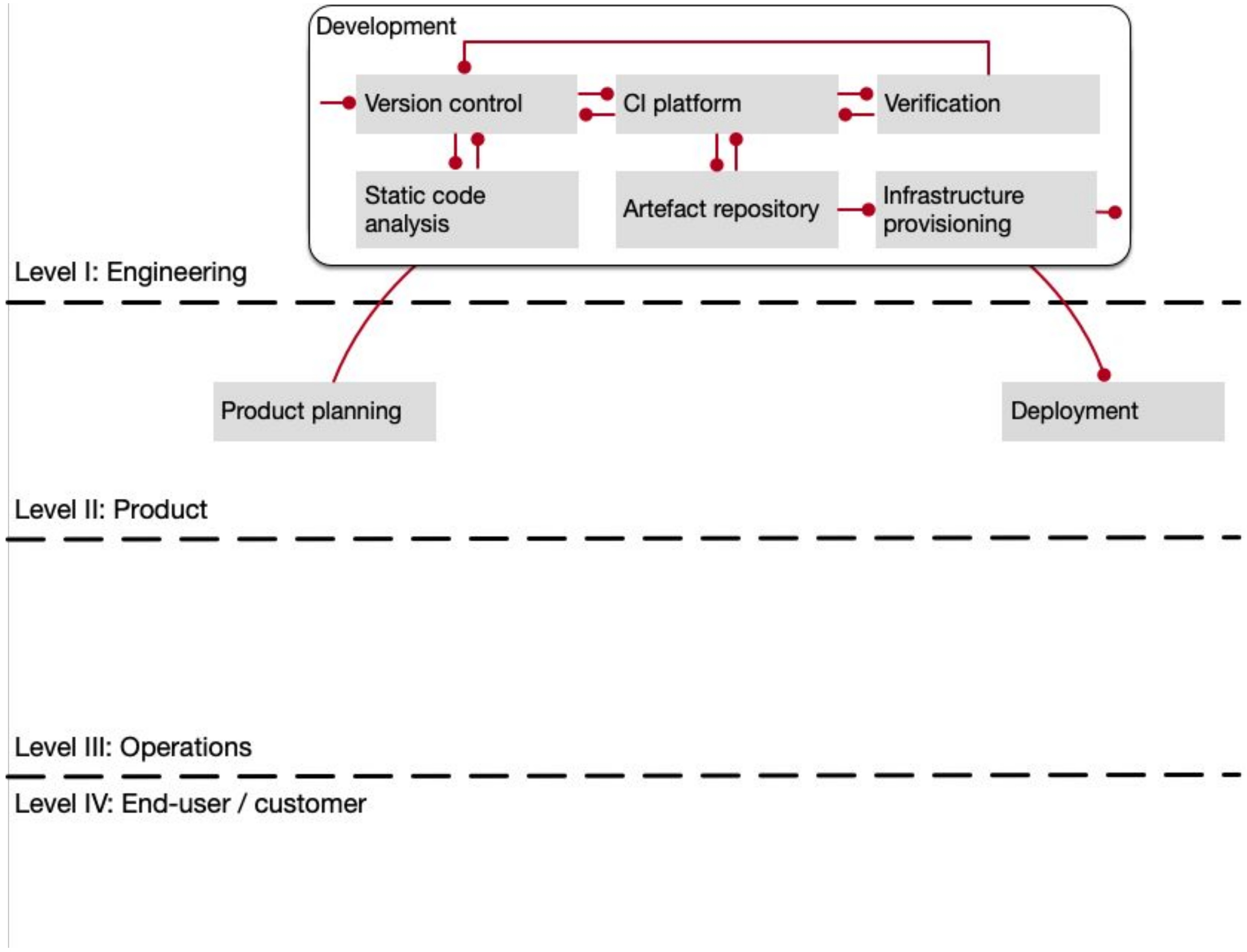
Level II: Product

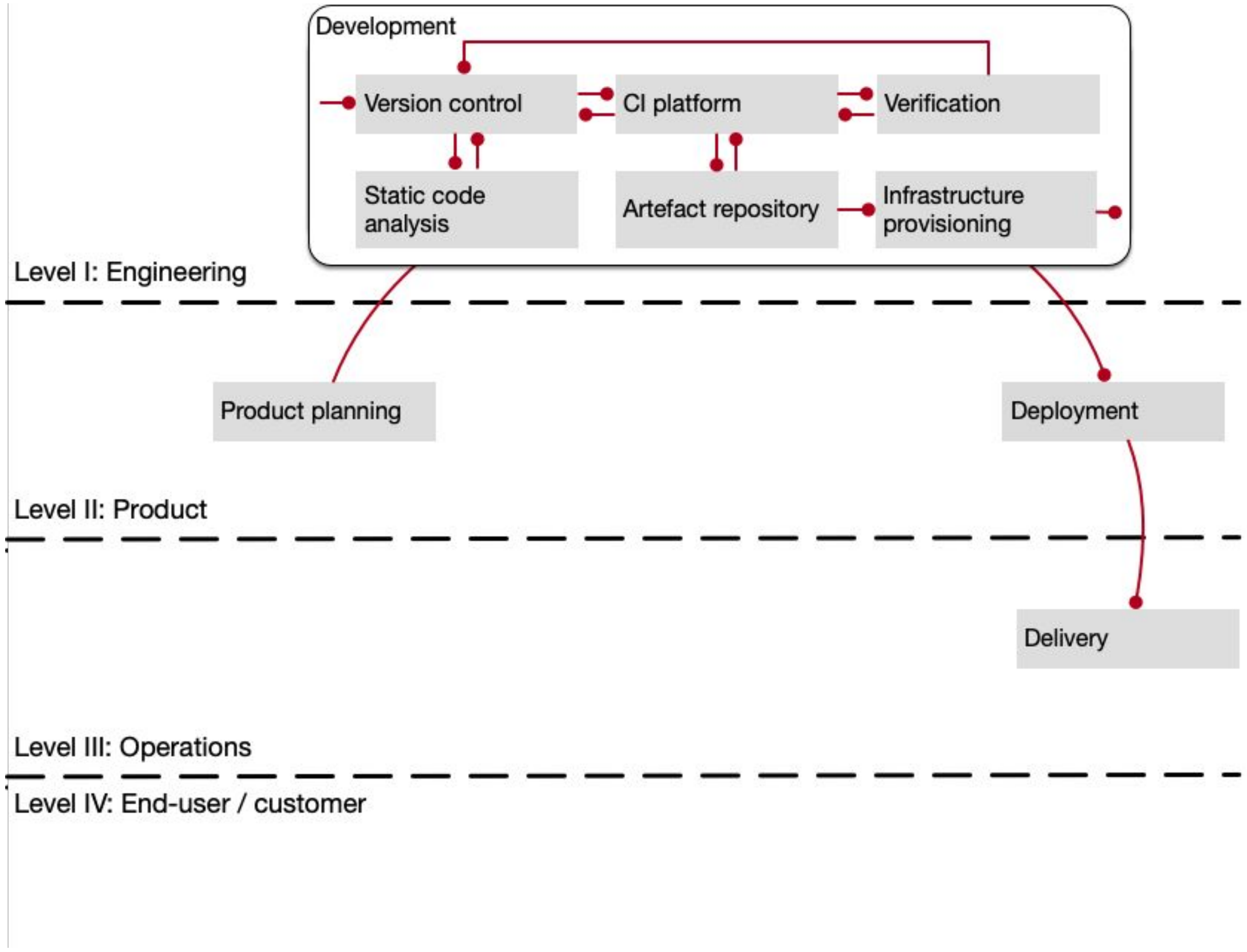
Level III: Operations

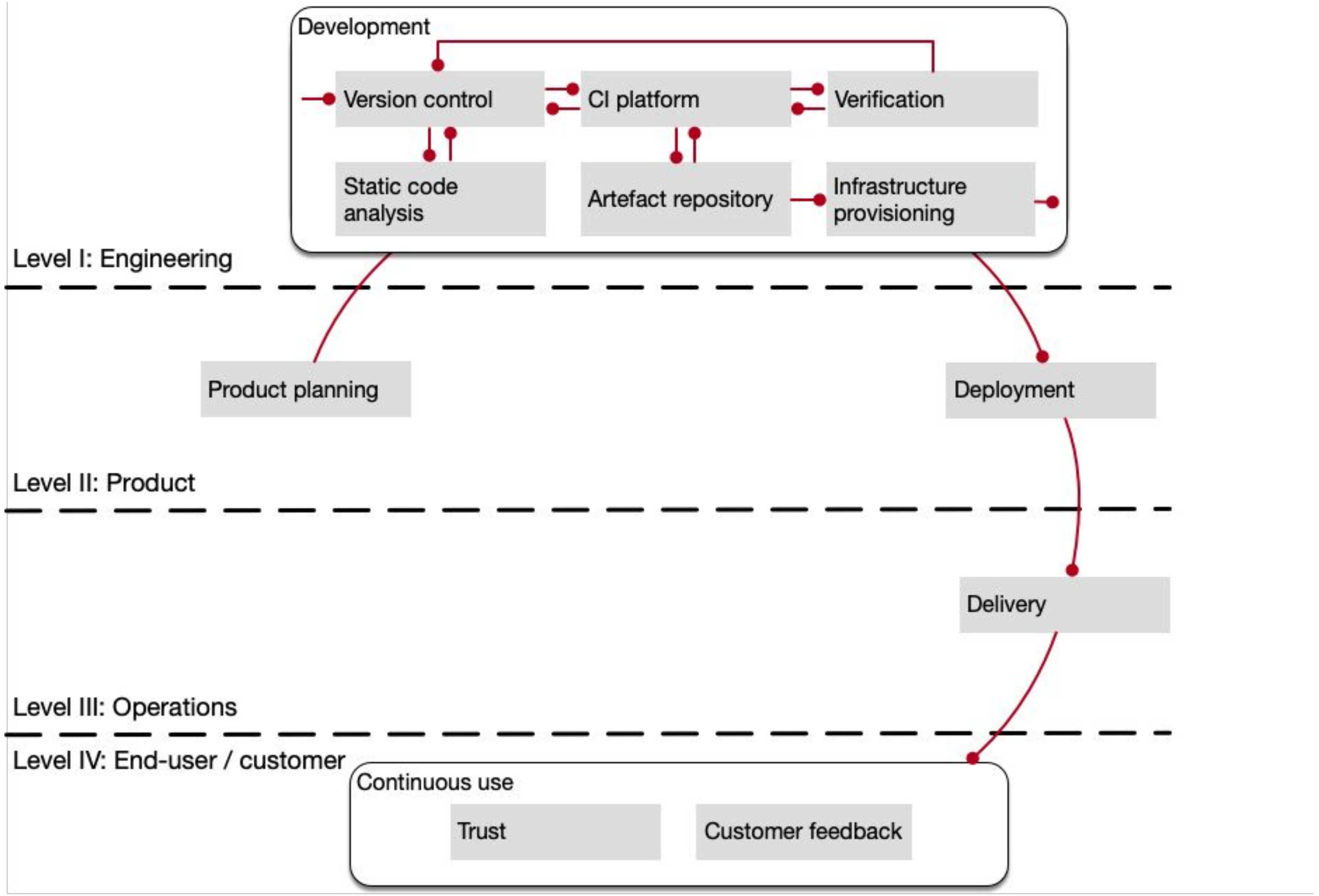
Level IV: End-user / customer

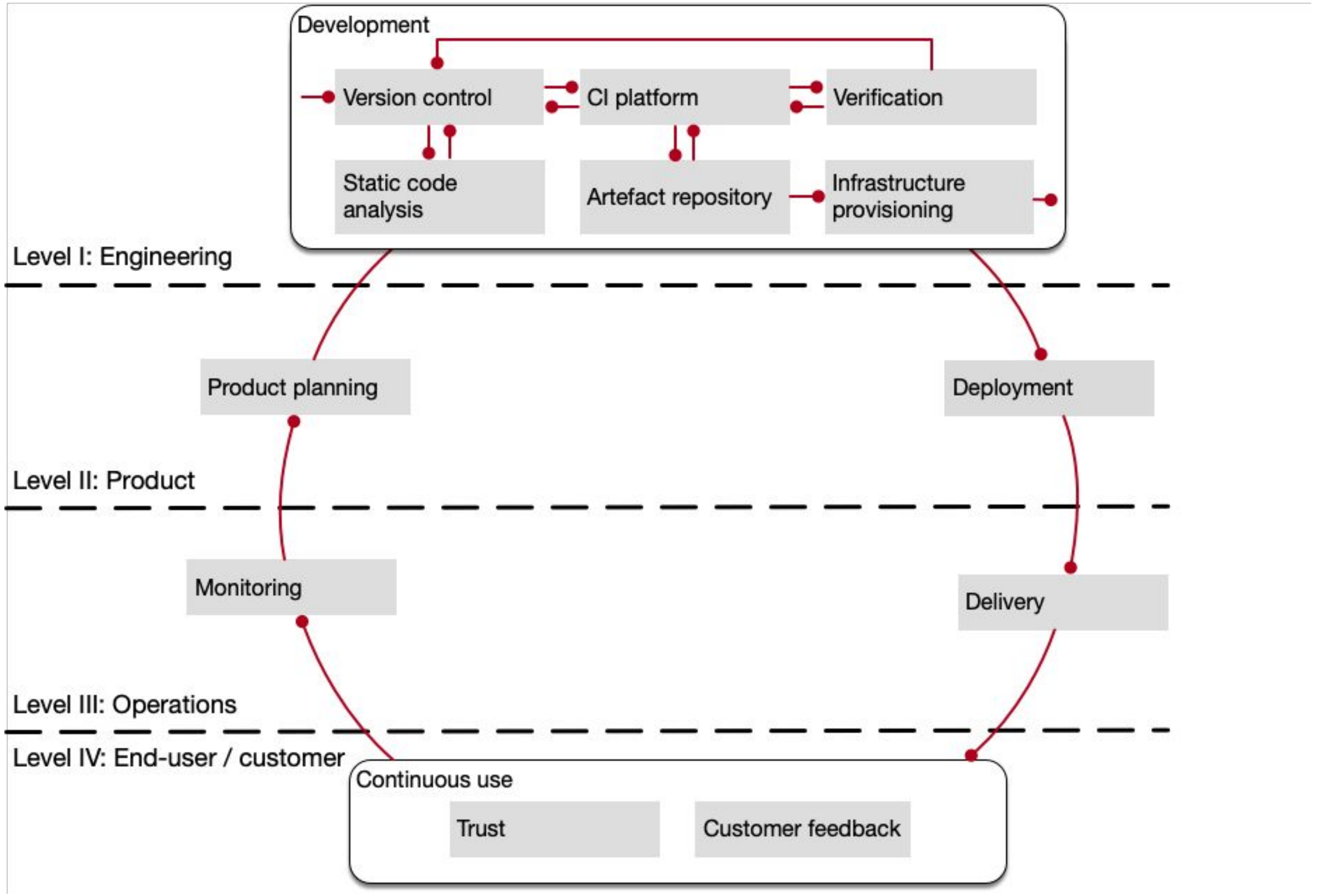

Level I: Engineering
Product planning
Level II: Product
Level III: Operations
Level IV: End-user / customer

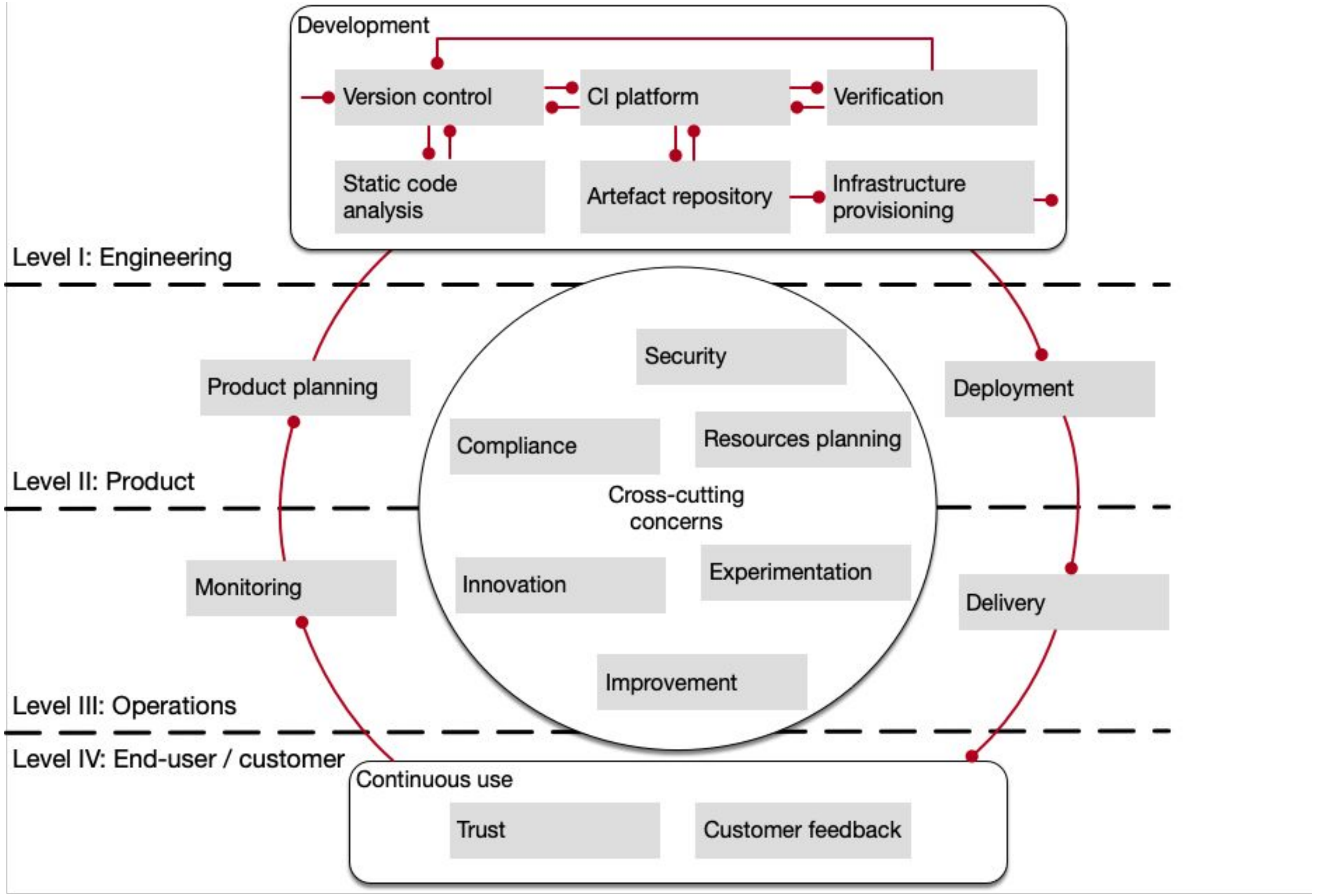

Development
Version control
CI platform
Verification
Static code analysis
Artefact repository
Infrastructure provisioning
Level I: Engineering
Product planning
Level II: Product
Level III: Operations
Level IV: End-user / customer


Development
Version control
CI platform
Verification
Static code analysis
Artefact repository
Infrastructure provisioning
Level I: Engineering
Product planning
Deployment
Level II: Product
Level III: Operations
Level IV: End-user / customer

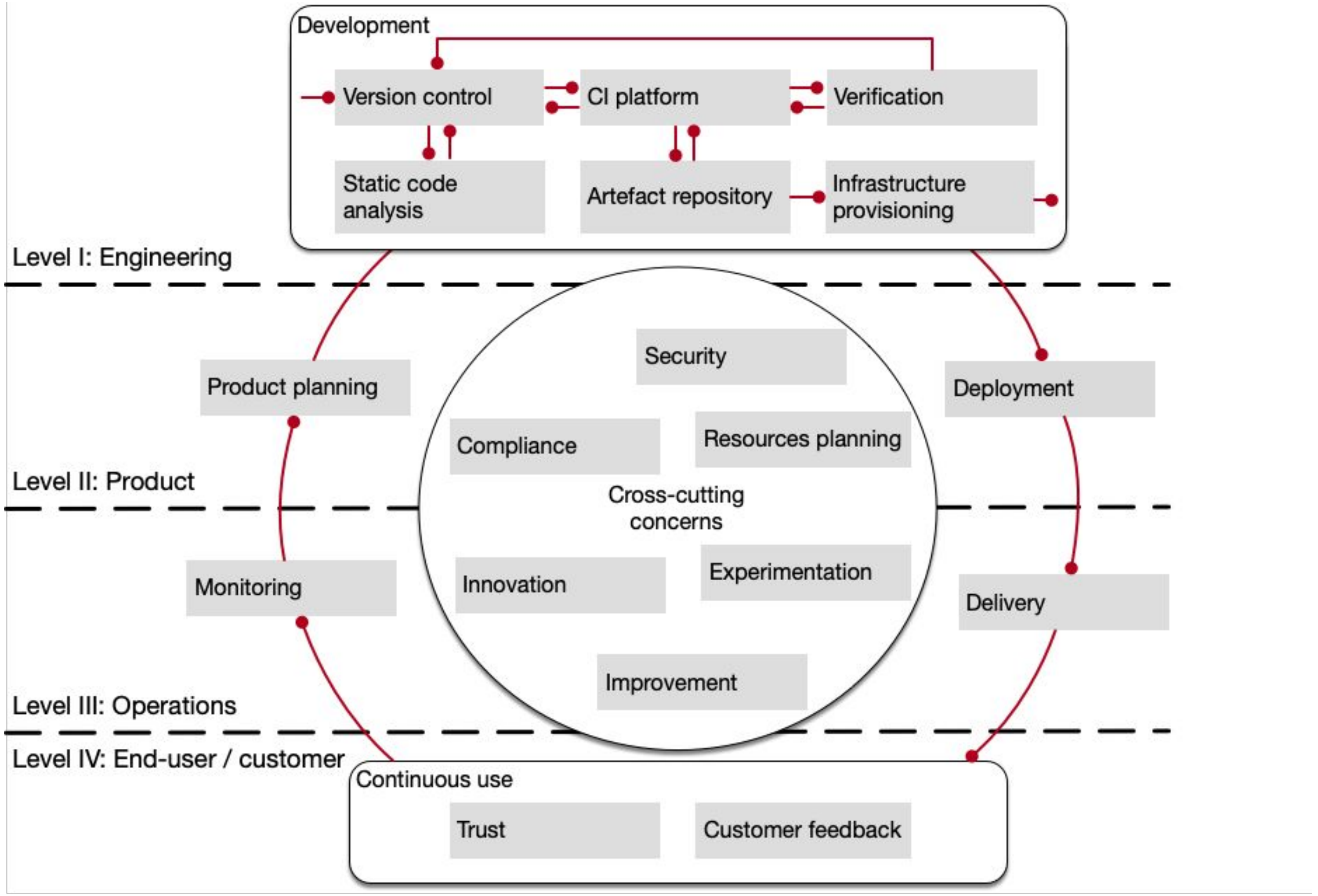

Development
Version control
CI platform
Verification
Static code analysis
Artefact repository
Infrastructure provisioning
Level I: Engineering
Product planning
Deployment
Level II: Product
Delivery
Level III: Operations
Level IV: End-user / customer

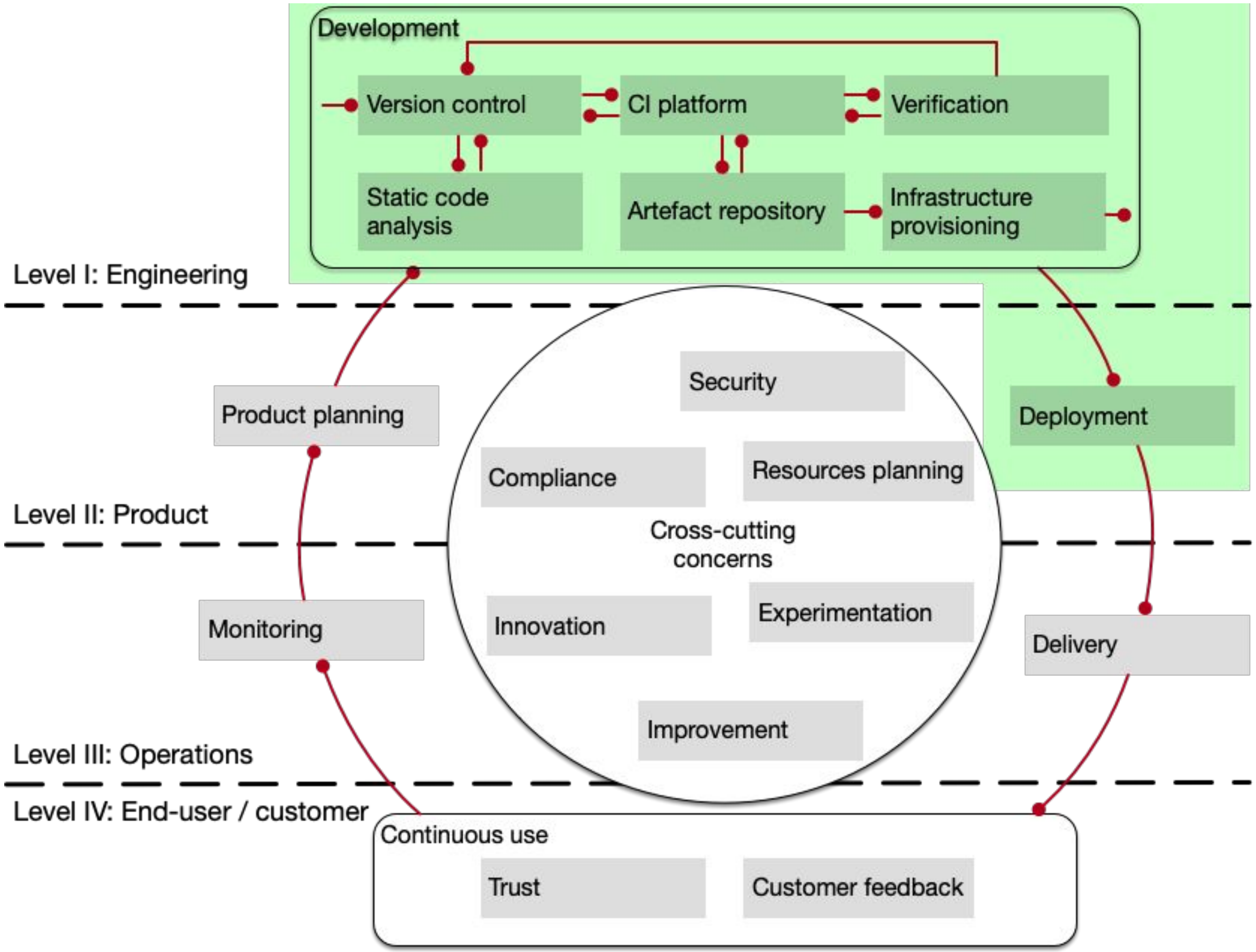

Development
Version control
CI platform
Verification
Static code analysis
Artefact repository
Infrastructure provisioning
Level I: Engineering
Product planning
Deployment
Level II: Product
Delivery
Level III: Operations
Level IV: End-user / customer
Continuous use
Trust
Customer feedback

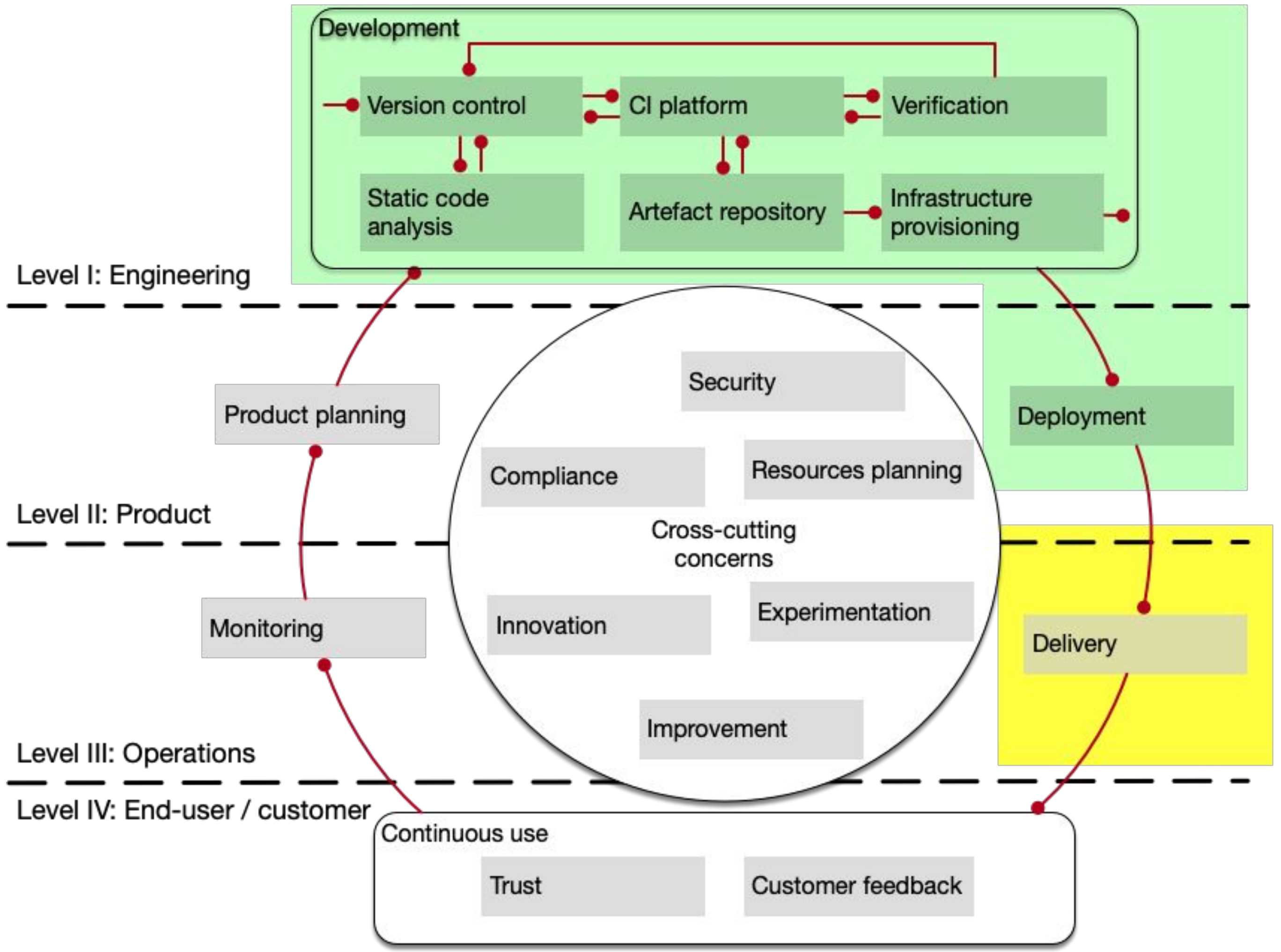

Development
Version control
CI platform
Verification
Static code analysis
Artefact repository
Infrastructure provisioning
Level I: Engineering
Product planning
Deployment
Level II: Product
Monitoring
Delivery
Level III: Operations
Level IV: End-user / customer
Continuous use
Trust
Customer feedback

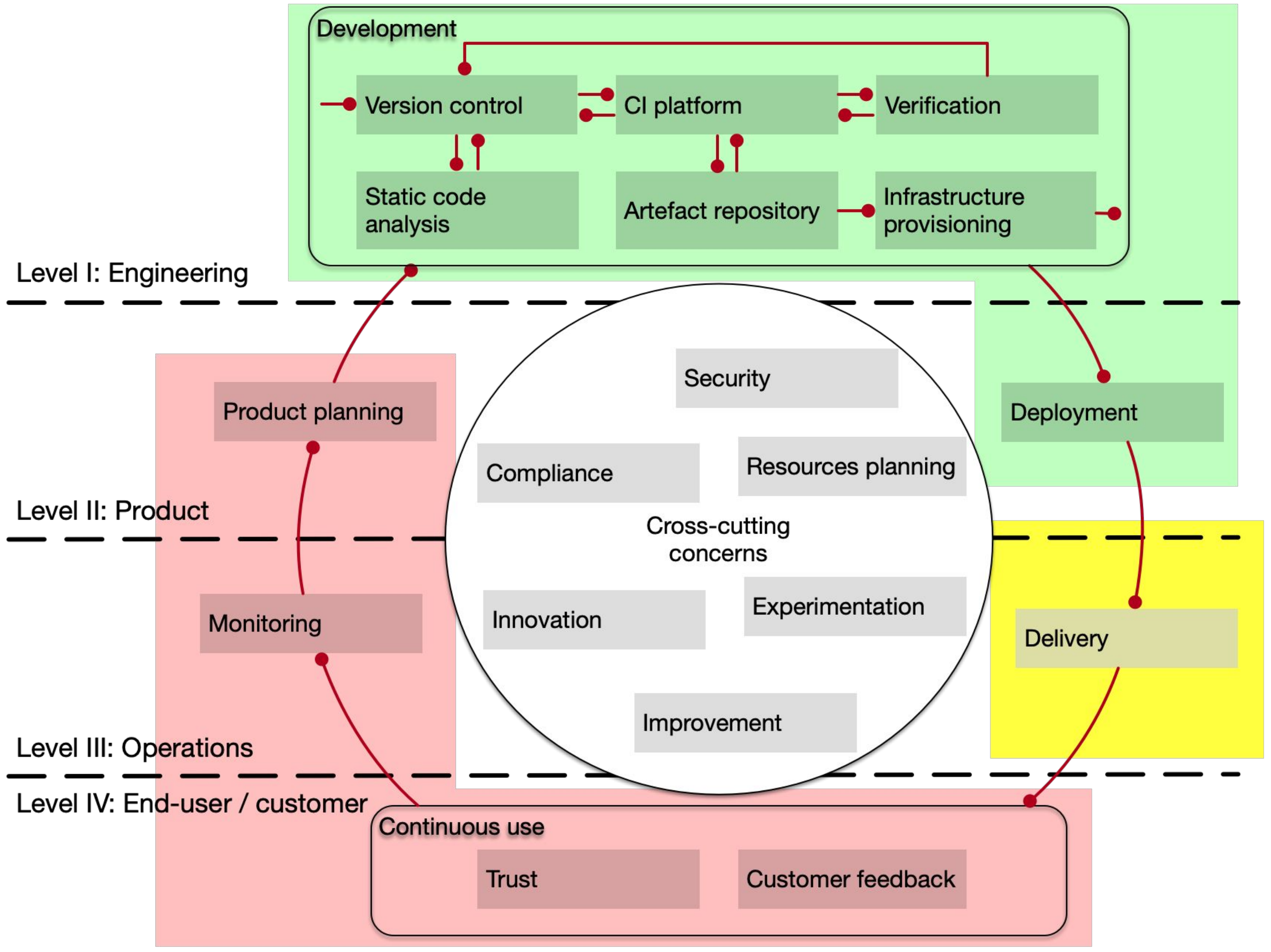

Development
Version control
CI platform
Verification
Static code analysis
Artefact repository
Infrastructure provisioning
Level I: Engineering
Product planning
Security
Deployment
Compliance
Resources planning
Level II: Product
Cross-cutting concerns
Monitoring
Innovation
Experimentation
Delivery
Improvement
Level III: Operations
Level IV: End-user / customer
Continuous use
Trust
Customer feedback

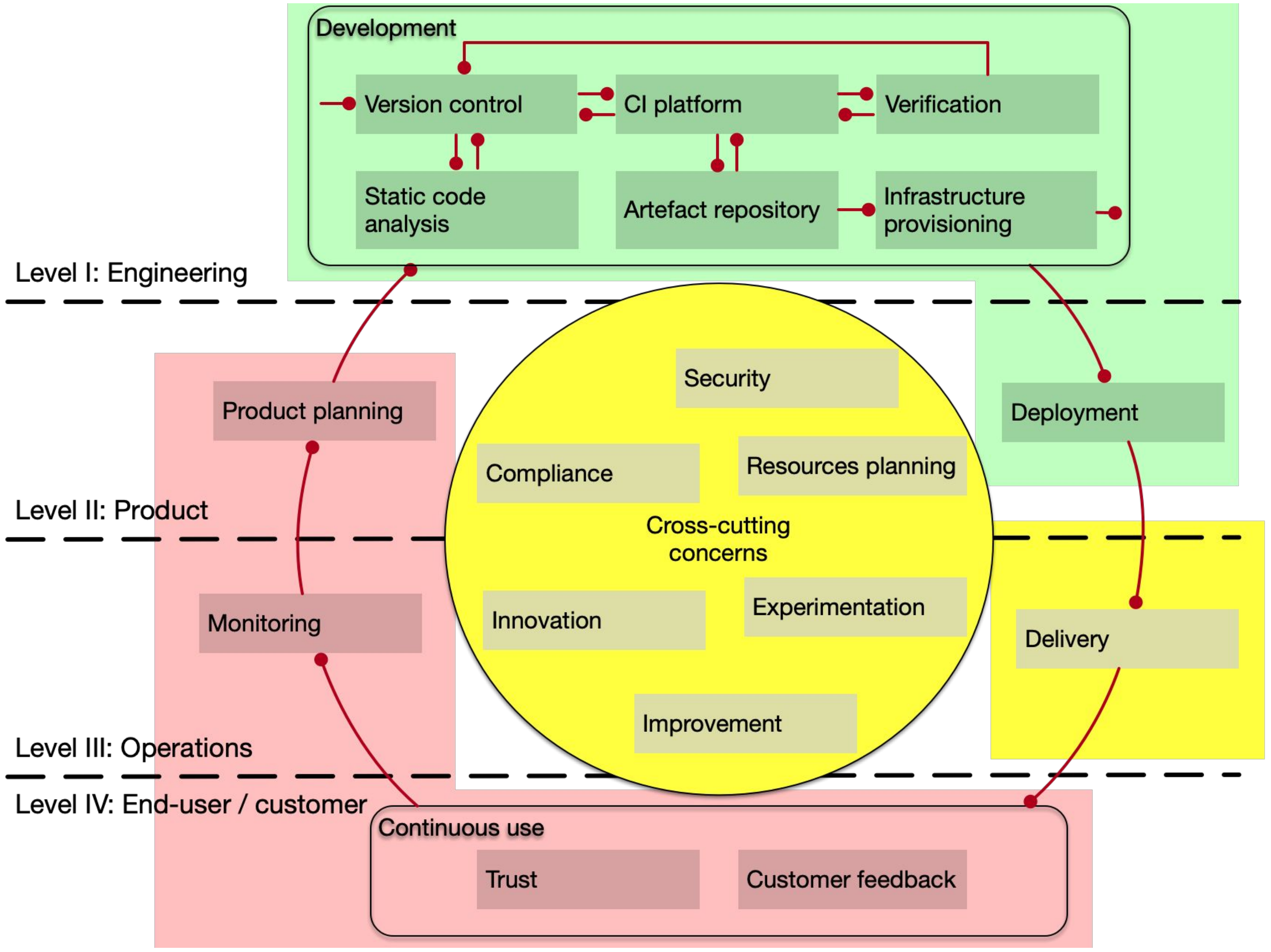
State-of-practice


Development
Version control
CI platform
Verification
Static code analysis
Artefact repository
Infrastructure provisioning
Level I: Engineering
Product planning
Security
Deployment
Compliance
Resources planning
Level II: Product
Cross-cutting concerns
Monitoring
Innovation
Experimentation
Delivery
Improvement
Level III: Operations
Level IV: End-user / customer
Continuous use
Trust
Customer feedback


Development
Version control
CI platform
Verification
Static code analysis
Artefact repository
Infrastructure provisioning
Level I: Engineering
Product planning
Deployment
Security
Compliance
Resources planning
Cross-cutting concerns
Level II: Product
Monitoring
Innovation
Experimentation
Delivery
Improvement
Level III: Operations
Level IV: End-user / customer
Continuous use
Trust
Customer feedback


Development
Version control
CI platform
Verification
Static code analysis
Artefact repository
Infrastructure provisioning
Level I: Engineering
Security
Product planning
Deployment
Compliance
Resources planning
Level II: Product
Cross-cutting concerns
Monitoring
Innovation
Experimentation
Delivery
Improvement
Level III: Operations
Level IV: End-user / customer
Continuous use
Trust
Customer feedback


Development
Version control
CI platform
Verification
Static code analysis
Artefact repository
Infrastructure provisioning
Level I: Engineering
Security
Product planning
Deployment
Compliance
Resources planning
Level II: Product
Cross-cutting concerns
Monitoring
Innovation
Experimentation
Delivery
Improvement
Level III: Operations
Level IV: End-user / customer
Continuous use
Trust
Customer feedback


Development
Version control
CI platform
Verification
Static code analysis
Artefact repository
Infrastructure provisioning
Level I: Engineering
Product planning
Security
Deployment
Compliance
Resources planning
Level II: Product
Cross-cutting concerns
Monitoring
Innovation
Experimentation
Delivery
Improvement
Level III: Operations
Level IV: End-user / customer
Continuous use
Trust
Customer feedback

Recurring challenges

BTH SERL Sweden LEADING SOFTWARE ENGINEERING

S.E.R.T reThink▪reThought

Challenge 1 - Determining the adoption goals

- **Superficial goals** like “speed”, “flexibility” and “efficiency” **are not useful** to drive any **systematic changes**
  - *E.g. efficiency* may have different meaning for different stakeholders, speed may be less relevant in slow-moving markets, continuous data sharing may be out of question for systems behind an air air-gap.
  - More {*flexible*|*frequent*|*efficient*|*speedy*} software delivery is implicitly assumed to be an improvement.
- Usually, improving one aspect happens at the expense of another. Few consider such tradeoffs.
- The goals should be **aligned**, measured and shared **across the whole organization**.

Challenge 2 - Dealing with Conway's law

- Software **architectures and processes** tend to follow the underlying **organizational structures**

- **Sub-optimal** hierarchical structures and **organizational conflict** lead to **functional silos** (e.g. strategy, product planning, R&D, QA, Sales, Operations) and **monolithic software.**

- Functional silos focus on their own “slice” and **optimize for their KPIs** without considering the whole picture.

- Any efforts to improve and automate the engineering process are **limited to a one silo**. Breaking silos and connecting the pipeline **requires changing** the organizational and software **structures**

Challenge 3 - Internal constraints

Driving changes in **large**, **old**, and “**stale**” **organizations** is inherently difficult

- **Culture and attitudes** play a significant role
- **Management** plays an important role
- Legacy products, processes, and structures **slow down changes**
- **Business models** may not be compatible
- **Politics** play a role

In principle, **no different** than driving any other **enterprise transformation**

Challenge 4 - External constraints

Having an **end-to-end** delivery pipeline **may not be possible** for all organizations and products due to:

- The lack of **incentives** to upgrade
- High **risk** of upgrading
- Downstream **dependencies** on products/vendors/processes
- Upstream **dependencies** on vendors/partners
- **Compliance** requirements
- Limited **control** over the software life-cycle

A sustained pace
of small improvements beats
occasional big changes

Contributions towards the future

- 💡 Implement structures and systems encouraging and supporting small improvements.
  - Collaboration, delegation, empowerment
  - Flexible processes
- 💡 Implement data-driven approaches for decision support
  - Measurable goals
  - Metrics, KPIs
  - Data pipelines, data warehouses
  - Broad access datasets, use of data and data analysis
  - Data literacy
- 💡 If *continuous X* is a relevant practice to improve towards a specific goal, implement it
- 💡 Measure and repeat