



# Volvo IT

DevOps development plan – kickoff

**200+**  
EMPLOYEES



DEVOPS PIONEER  
SINCE  
**2007**



CURRENT  
TURNOVER  
**10 M€**



OPERATES FROM  
**HELSINKI**  
**TAMPERE**  
**COPENHAGEN**  
**STOCKHOLM**  
**BEIJING**

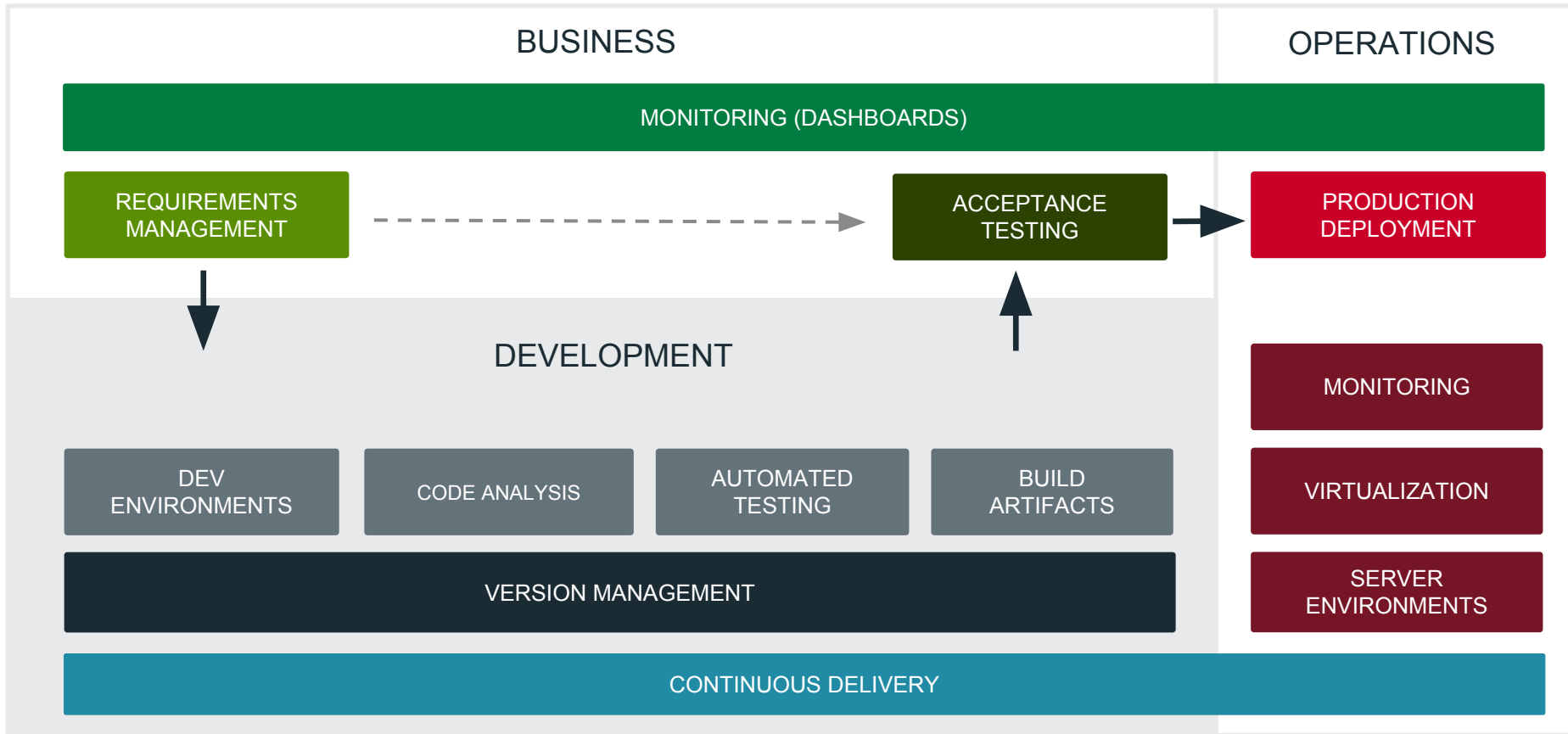
# ABOUT EFICODE



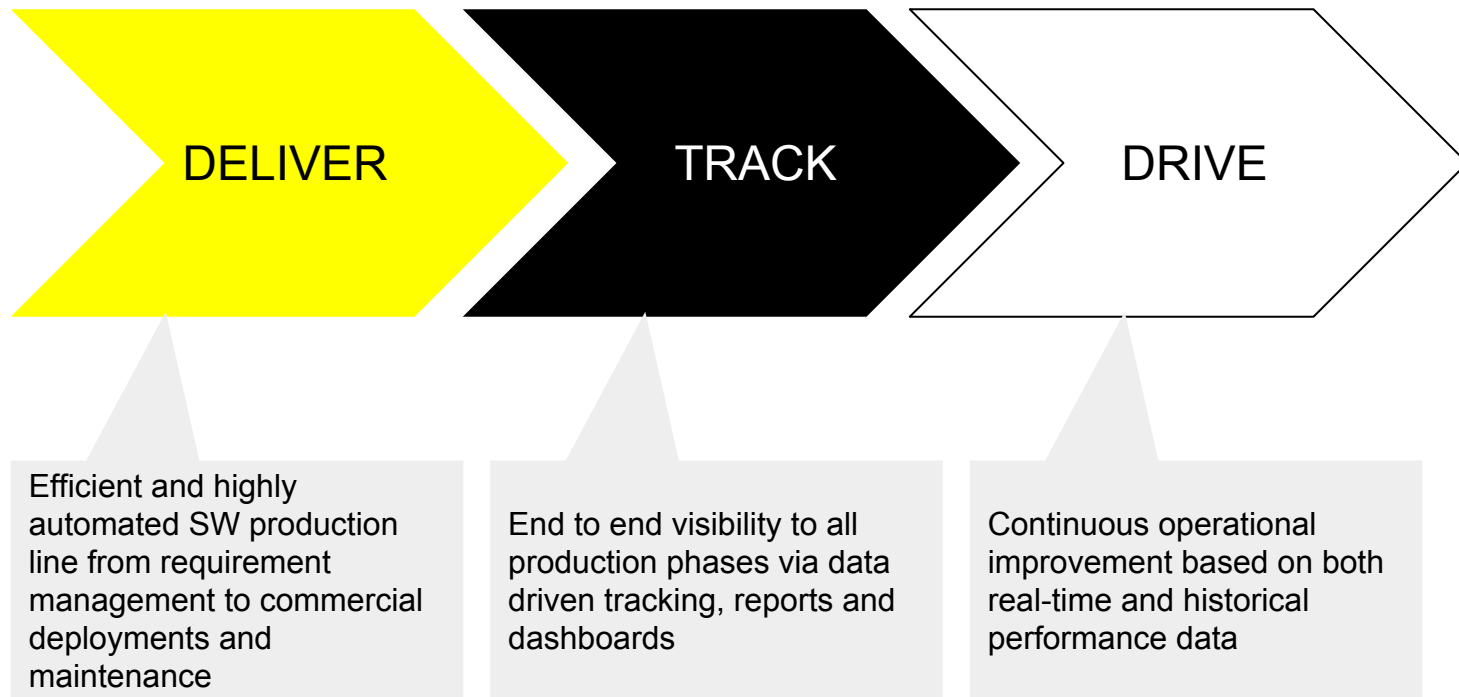
# CODE FOR GROWTH

From ideation to building.  
From release to the future.

# DEVOPS DEVELOPMENT MODEL



# DEVOPS TRANSFORMATION JOURNEY





## **VOLVO IT KPIs:**

Reduce Lead-time  
Increase Quality



Pilot Identification & Buy-In  
Baseline and Impact analysis  
Build Implementation Plan

# STUDY PROCESS





# PROJECT REPORT

- DevOps maturity model benchmark
- Automation maturity benchmark
- Key findings from the perspective of the DevOps
- Improvement suggestions for the identified areas
- Roadmap for suggested improvements



# DEVOPS MATURITY - **XX** /100

eficode

## LEADERSHIP

## ORGANIZATION AND CULTURE

## ENVIRONMENTS AND RELEASE

## BUILDS AND CONTINUOUS INTEGRATION

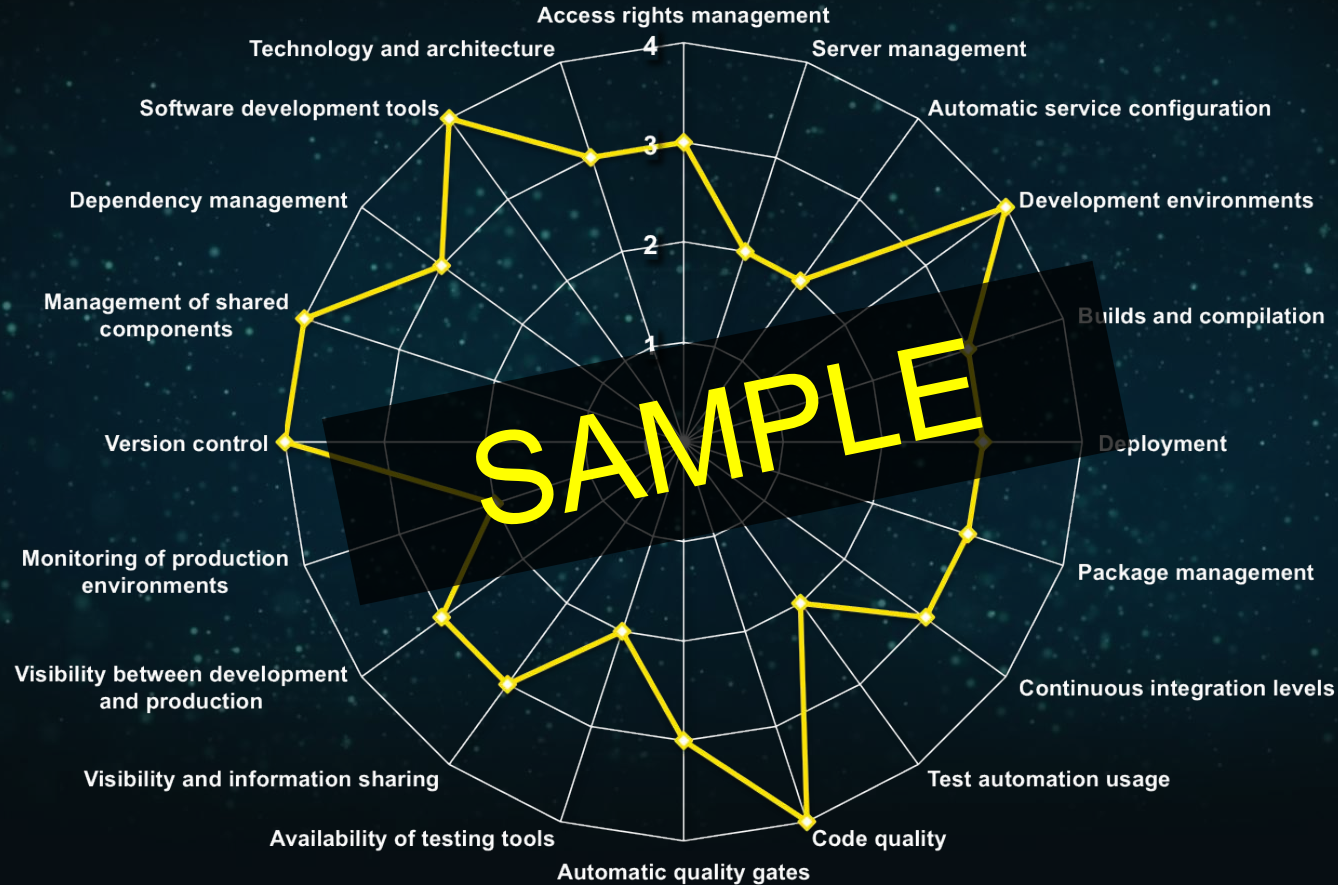
## QUALITY ASSURANCE

## VISIBILITY AND REPORTING

## TECHNOLOGIES AND ARCHITECTURE

001	002	003	004
Development operations have been separated from the business knowledge. Starting a new development project is laborious.	Starting new development projects is agile, and there are practices in place for steering the project.	New projects can be connected to organization's strategic targets. Starting a new pilot project is easy.	Real-time metrics supporting decision making and tracking the completion of strategic targets are available.
Design, development and quality assurance are separate from each other. Communication is primarily in writing.	Work is conducted in teams but development and quality assurance are separate from each other.	The teams work independently. They have total liability for the development and quality assurance of features.	The teams communicate with each other regularly and work together to improve their practices. Communication with the IT operations is continuous.
Products are environment-specific and they are compiled manually. Environments are installed and configured manually.	The system is divided into parts and the compiling environment is known. Some releases are automated.	Environments can be installed and configured automatically. Build and release processes are automated.	Releases may be conducted automatically and continuously. Migration and recovery processes work as expected.
Product integration is automatic, but configuration and deployment are controlled manually. No artifact or change logs management.	The process starts and specifically after every change. Tools are shared. Integration does not involve testing.	Integration covers the entire product and it is connected to acceptance testing. Dependencies are known and managed.	Build and integration processes are continuously improved based on collected metrics with aim to speed up the feedback cycle and improve visibility.
Quality assurance is conducted completely by hand and primarily after development.	Unit testing or static code analysis is in place for some parts of the product.	Features visible to the end users are covered with automatic tests. Testers participate in the development process.	Acceptance tests present system requirements clearly and guide the development of the system as much as possible.
Reports are made by hand when necessary.	Code integration, unit testing and code analysis are visible to the team.	The status of requirements can be monitored in real time in relation to tests and released features.	Real-time metrics are automatically collected from the product development process and used as a basis for improvement.
Technologies and tools are obsolete or are not fit for current requirements.	Technologies are growing old and the architecture is only partially adaptive or the interfaces are lacking.	Technologies are modern or well supported. The interfaces are well documented and exist for all key functionalities.	The architecture and technologies are optimal and enable reaching business targets efficiently.

# AUTOMATION MATURITY - XX /100





**THANK YOU**