Deloitte.

DevOps Point of View

An Enterprise Architecture perspective

Amsterdam, 2022

Management summary

"It is not the strongest of the species that survive, nor the most intelligent, but the one most responsive to change."[1]

Setting the scene

In the current world of IT and the development of IT-related products or services, companies from enterprise level to smaller sizes are starting to use the DevOps processes and methods as a part of their day-to-day organization process.

The goal is to reduce the time involved in all the software development phases, to achieve greater application stability and faster development cycles.

However not only on the technical side of the organization is DevOps changing the playing field, also an organizational change that involves merging development and operations teams is required with a hint of cultural changes.

And last, but not least, the skillset of all people involved is changing.

Goal of this Point of View

This point of view aims to create awareness around the transformation towards the DevOps way-of-working, to help gain understanding what DevOps is, why you need it and what is needed to implement DevOps.

An Enterprise Architecture perspective

Even though it is DevOps from an Enterprise Architecture perspective, this material has been gathered from our experiences with customers, combined with knowledge from subject matter experts and theory from within and outside Deloitte.

Targeted audience

It is specifically for the people within Deloitte that want to use this as an accelerator for conversations & proposals and to get in contact with the people who have performed these type of projects.

By all means, it is a deck that can be shared within Deloitte and with our customers to provide a more holistic view.

[1] Charles Darwin

© 2022 Deloitte The Netherlands

DevOps practitioners

For questions or remarks, feel free to reach out to our DevOps practitioners

Mark Maijs

DevOps Overall Lead

Director
Technology, Strategy & Transformation
Deloitte Consulting
+31 6 83 89 02 14
mmaijs@deloitte.nl

Donald Pondman

DevOps Transformation

Manager
Technology, Strategy & Transformation
Deloitte Consulting
+31 6 83 33 02 56
dpondman@deloitte.nl

Sebastiaan Teerenstra

DevOps Process & People

Senior Consultant
Technology, Strategy & Transformation
Deloitte Consulting
+31 6 22 98 09 09
steerenstra@deloitte.nl

Marlies Quekel

DevOps Organization & Team Setup

Consultant
Technology, Strategy & Transformation
Deloitte Consulting
+31 6 50 07 02 16
mquekel@deloitte.nl

Lianne Veenhuis

Test Management

Business Analyst
Technology, Strategy & Transformation
Deloitte Consulting
+31 6 156 904 09
lveenhuis@deloitte.nl

© 2022 Deloitte The Netherlands

Contents

What is DevOps? 5

Why do I need DevOps? 9

Where is DevOps applicable? 12

What is needed for DevOps to work? 14

- People 16
- Process 24
- Technology 27
- Operating Model 33

How do I implement DevOps? 38

Deloitte Accelerators 42

Client Examples 46

© 2022 Deloitte The Netherlands

What is DevOps?

© 2022 Deloitte The Netherlands

What is DevOps?

DevOps is a new way-of-working that improves value delivery for the customer and enables benefits for both development and operations

Definition

DevOps is a new approach to optimize and manage end-to-end service delivery and operations. It applies a set of principles to transform the entire software delivery lifecycle to introduce new practices enabled by technology

Goal

DevOps primary goal is to improve the flow from an idea towards value for the customer, enabled by an environment in which multidisciplinary teams work collaboratively to continuously deliver high quality solutions, in a faster pace, that qualify for operations

DevOps principles

- Culture of shared responsibility and collaboration
- End-to-end ownership of services
- Multi-disciplinary teams
- Incremental value delivery
- Flow optimization in the delivery process
- Automate (almost) everything
- Measurement of everything
- Continuous improvement

Software Delivery Lifecycle

New DevOps practices:

- Continuous Integration
- Continuous Testing
- Continuous Delivery
- Continuous Operations

Applying DevOps principles to the SDLC lead to new practices that benefit both Development and Operations

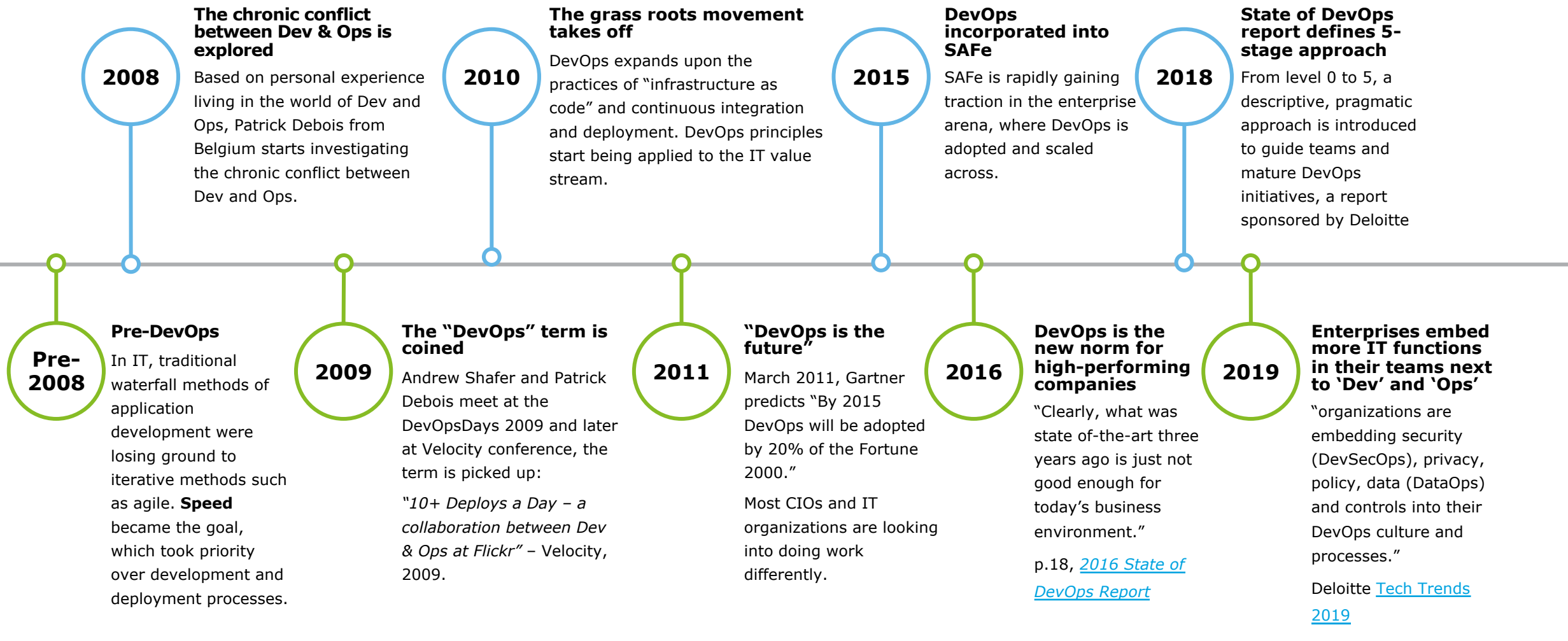
Benefits

- Increases the frequency and quality of deployments and releases
- Improves innovation and risk-taking
- Realizes faster time to market
- Improves solution quality and operational reliability
- Improves the Mean Time to Recover (MTTR)

© 2022 Deloitte The Netherlands

The History of DevOps

DevOps is becoming the norm in software delivery and is increasingly being adopted & matured across enterprises, becoming the new best practice



© 2022 Deloitte The Netherlands

DevOps practices

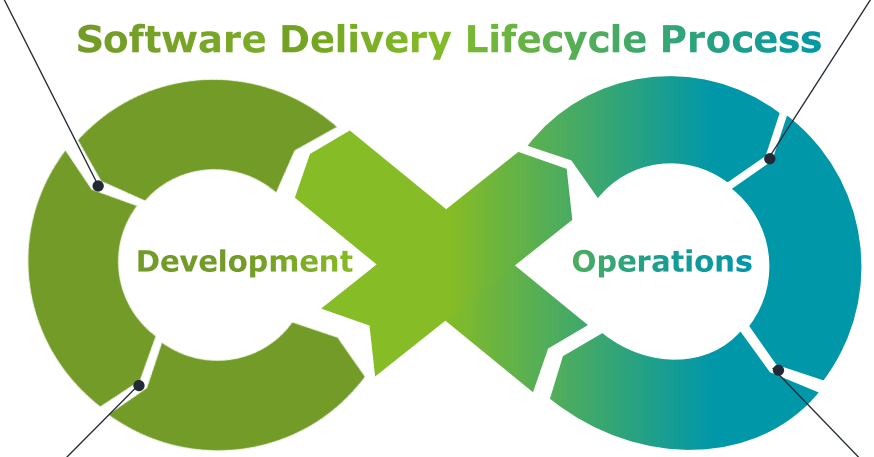
DevOps practices apply continuous automation cycles throughout software development and operations processes

Continuous Integration

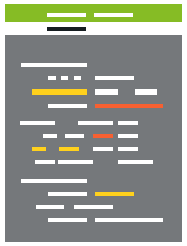
the streamlining of internal development by integrating code into a shared repository several times a day. Each check in is then verified by an automated build, allowing teams to detect problems early in the cycle

Continuous Delivery

is the process of delivering code that is production ready and is kept in an always releasable state, so it can be deployed (automatically) to production at any given time based on business needs



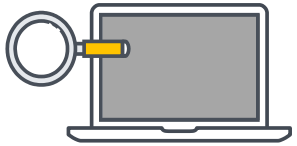
Continuous Testing



automating and integrating tests into the software delivery chain, and automatically executing those tests against each build of the code base

Continuous Operations

is proactively managing the solution based on feedback loops. Monitoring and telemetry become part of the backlog. Processes such as patching also fall under this practice



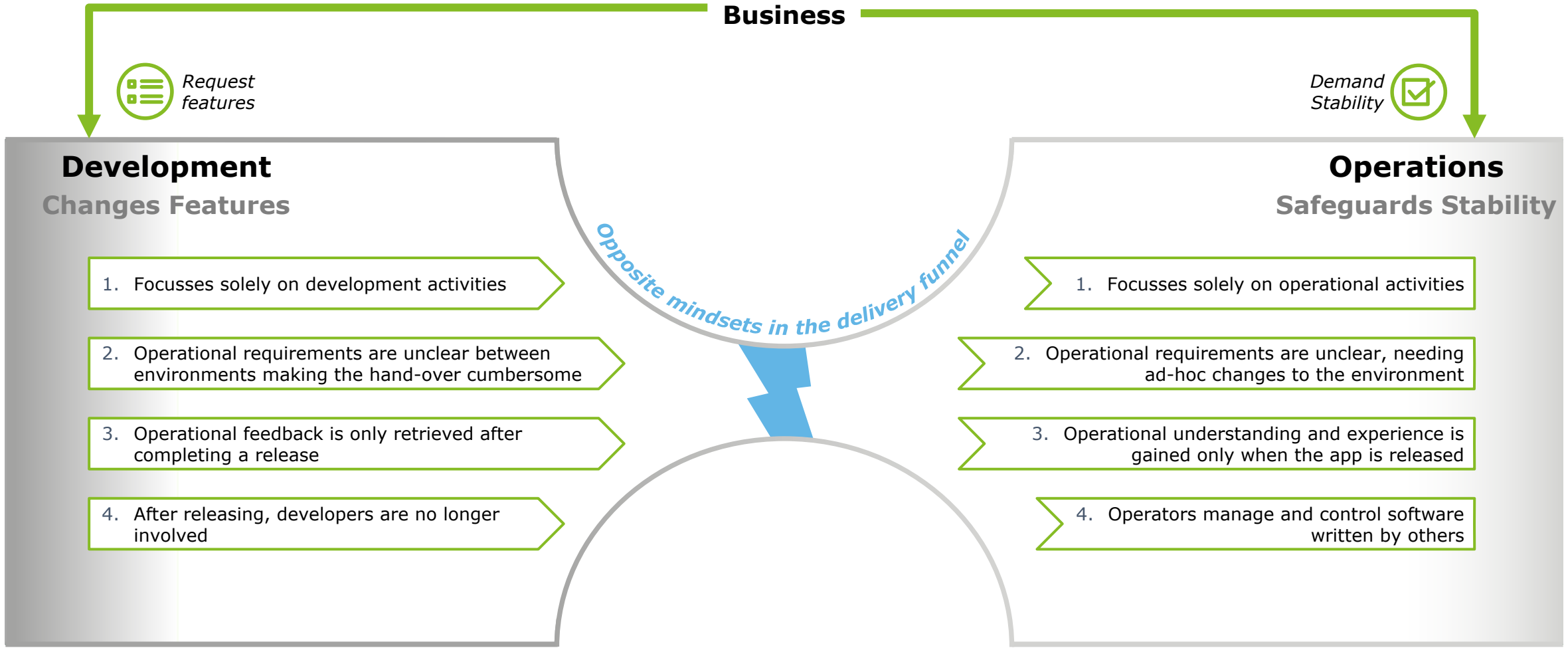
© 2022 Deloitte The Netherlands

Why do I need DevOps?

© 2022 Deloitte The Netherlands

Traditional function separation for Development and Operations

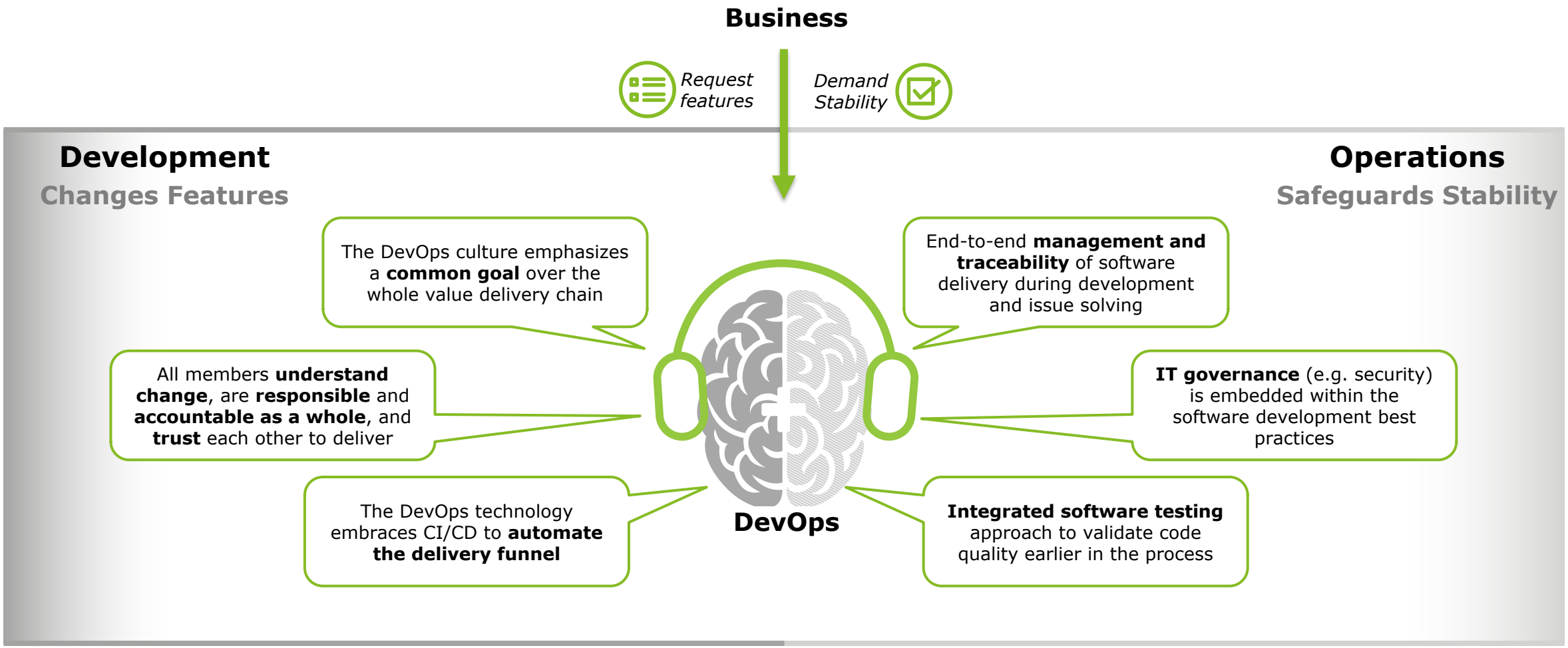
Prior to DevOps change release frequency was low, Development and Operations worked separately to serve business demands, having completely opposite mindsets



© 2022 Deloitte The Netherlands

DevOps unifies the mindset of Development and Operations

Today business wants to release on demand. With DevOps, both functions continuously collaborate to align business demands within the software delivery lifecycle



© 2022 Deloitte The Netherlands

Where is DevOps applicable?

© 2022 Deloitte The Netherlands

Indicators for DevOps Applicability

Several indicators help to determine if DevOps is applicable for your organization

DevOps is Applicable	Indicator	DevOps is Not Applicable
Management trusts delivery teams to work autonomously and only shares a product vision	Leadership style	Management requires direct involvement in the delivery process and makes all decisions
Multiple teams are responsible to manage the end-to-end lifecycle of a single product	Team composition	Product or service delivery does not require a multi disciplinary (cross-functional) team
Environments where IT solutions are changing rapidly	Change rate	Environments where IT solutions have low change rate
An incremental delivery process that focuses on early value delivery	Delivery process	Your delivery process has many sequential constraints, where outputs equal required inputs for consecutive process steps
Desired product end-state is unknown, changing business requirements give guidance on steering development	Business uncertainty	Desired product end-state is known and business requirements do not often change
People have great affinity with software and technology and are not change averse	Change willingness	People have no affinity for new technologies, and are change averse
Product is software that could be delivered as-a-service	Product type	Products are tangible, typically consisting of semi-finished products provisioned by multiple partners that don't have a direct relation with each other

© 2022 Deloitte The Netherlands

What is needed for DevOps to work?

© 2022 Deloitte The Netherlands

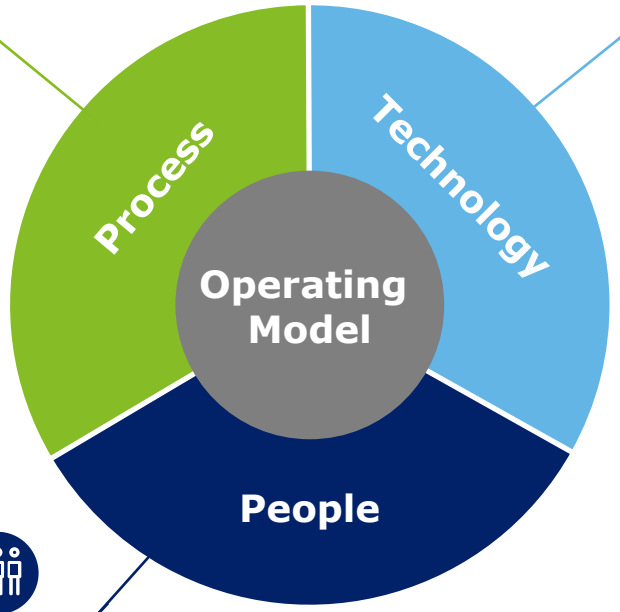
DevOps dimensions

The DevOps operating model is structured along People, Process and Technology, each dimension is necessary for successful DevOps

Process

Establish standardized interconnected process in the software development (and operation) lifecycle

- Continuous Everything: integration, delivery, testing, monitoring, release management and planning
- Continuous Integration and Continuous Delivery are key to build quality into DevOps processes
- Establish interconnected processes across all phases of development and operations for consistent and predictable deployments



Technology

Improve toolset to support the delivery and automation of the process specifically to accelerate software delivery activities

- Container based delivery and immutable infrastructure blocks
- Leverage the vast DevOps tooling landscape to automate and support Continuous Integration and Continuous Delivery and minimize user intervention
- Support of dynamic environment configuration to help remediate the current bottleneck in testing environment availability

People

Establish a DevOps Organization & Culture with cross-functional teams that are open and trustful

- T-shaped employees
- Foster continuous learning and development to build cross functional capabilities and a mindset open to continuous change
- Transformational leadership & balanced metrics to drive DevOps culture
- Open and transparent communication enable feedback and short learning cycles

Operating Model

- People, Process and Technology combined in a governance model for the DevOps way-of-working
- Teams deliver services end-to-end in the DevOps Target Operating Model

© 2022 Deloitte The Netherlands

People

© 2022 Deloitte The Netherlands

DevOps Organization & Culture

The principles of the DevOps way-of-working have extensive implications on the organization structure, as well as on the culture of the workforce

DevOps principle	Implication for Organization & Culture
Culture of shared responsibility and collaboration	Teams are accountable for progress and output, not an individual team member. Team setup is persistent and co-located to improve collaboration and performance.
End-to-end ownership of services	Team resources are allocated by services instead of organizational functions. Teams take end-to-end accountability and responsibility (vertically integrated) for the delivery of a service.
Multi-disciplinary autonomous teams	Teams are setup vertically, end-to-end responsible for the whole lifecycle of a product. It contains balanced T-shaped skilled personnel from various domains (cross-functional) to achieve its targets.
Incremental value delivery	Work is broken down into small pieces to continuously deliver value to the business using iterative and frequent releases.
Flow optimization in the delivery process	Elimination of waste, shift left and limit work in progress optimizes the flow in the delivery process. Teams test as early and as often as possible, minimize handoffs and maximize checkpoints to reduce dependencies and risks.
Automate (almost) everything	Tools automate as many tasks and process steps as possible in the delivery process to drastically reduce time, effort, and risk of human errors.
Measurement of everything	Everything is monitored and measured by a balanced metric system focused on the speed and stability of service delivery.
Continuous improvement	Teams organize retrospectives, (automated) feedback loops, and touchpoints with the business in order to continuously improve their delivery and way-of-working.

© 2022 Deloitte The Netherlands

DevOps principles are the starting point for an organization structure

Based on DevOps principles, an organization allocates resources by service instead of functions to enable end-to-end ownership and increase agility within teams

DevOps principles

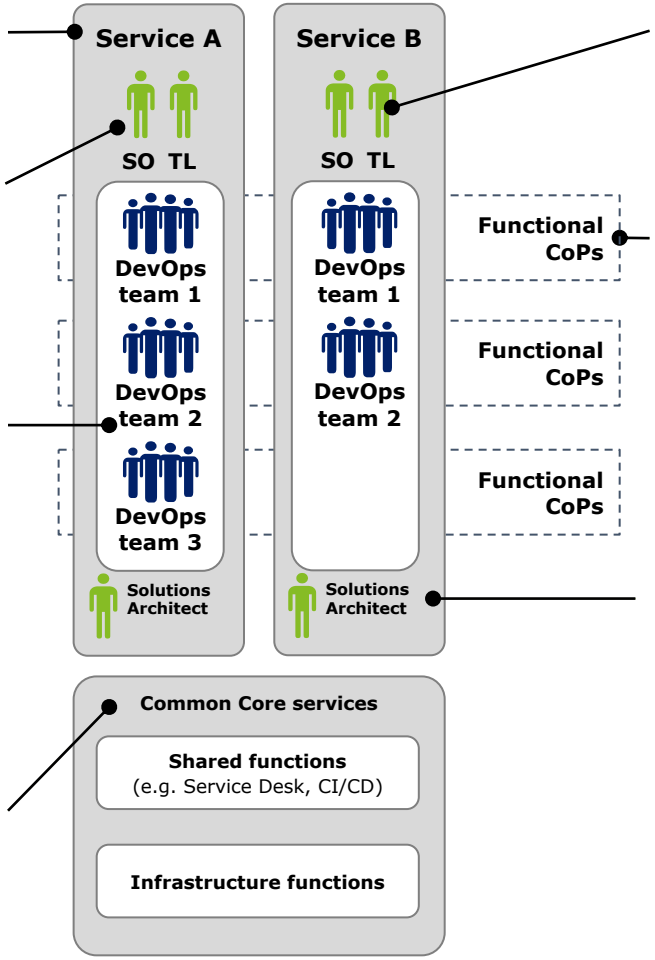
- Culture of shared responsibility and collaboration
- End-to-end ownership of services
- Multi-disciplinary autonomous teams
- Incremental value delivery
- Flow optimization in the delivery process
- Automate (almost) everything
- Measurement of everything
- Continuous improvement

Product/Service multi-disciplinary team(s) are aligned to a particular service

Service Owner is leading delivery and manages the development and operations of the service

Cross Functional DevOps team is a long-standing, fully-allocated, cross functional team that is end-to-end responsible for (a module of) the delivered product

Common Core services are shared services provided, preferably through self-service portals, by teams of the technology organization to be used by the different DevOps teams



Tech Lead is a product team member with extensive technical development experience who can lead the Product Team in the execution of its work

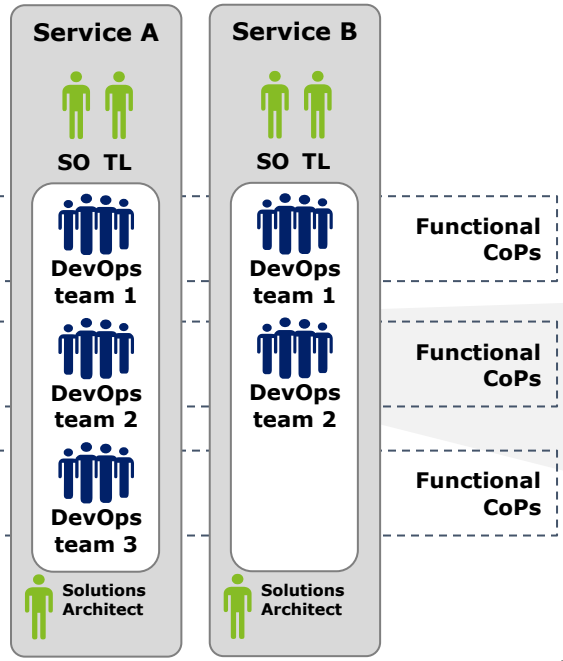
Functional Communities of Practices are the knowledge sharing and communication glue that keeps functional expertise (e.g. QA, development, testing etc.) together

Solution Architect is a product team member who governs architecture, design and implementation while enforcing architecture standards and guidelines

© 2022 Deloitte The Netherlands

Key DevOps roles and responsibilities

Several key roles should be represented in a cross functional DevOps team; a team member with a T-shaped profile can fulfill more than one role



The **Product Owner** leads the DevOps team and directs development to maximize business value

The **Business Analyst** engages the business for requirements, helps defining features, user stories & test cases, and validates designs

The **Developer** writes and verifies code, fixes bugs, executes patch management, maintains asset and configuration repository and functions as 2nd line support

The **Test Engineer** creates and executes test scripts, automates tests, supports usability testing & UAT, and manages test environments and test data

The **DevOps Engineer*** configures, maintains and monitors the CI/CD toolset to accelerate and de-risk application delivery

The **Operations Specialist** executes day-to-day technology operations (functional maintenance), monitors technology operations, performs Problem Management, manages change processes (Approves/Rejects)

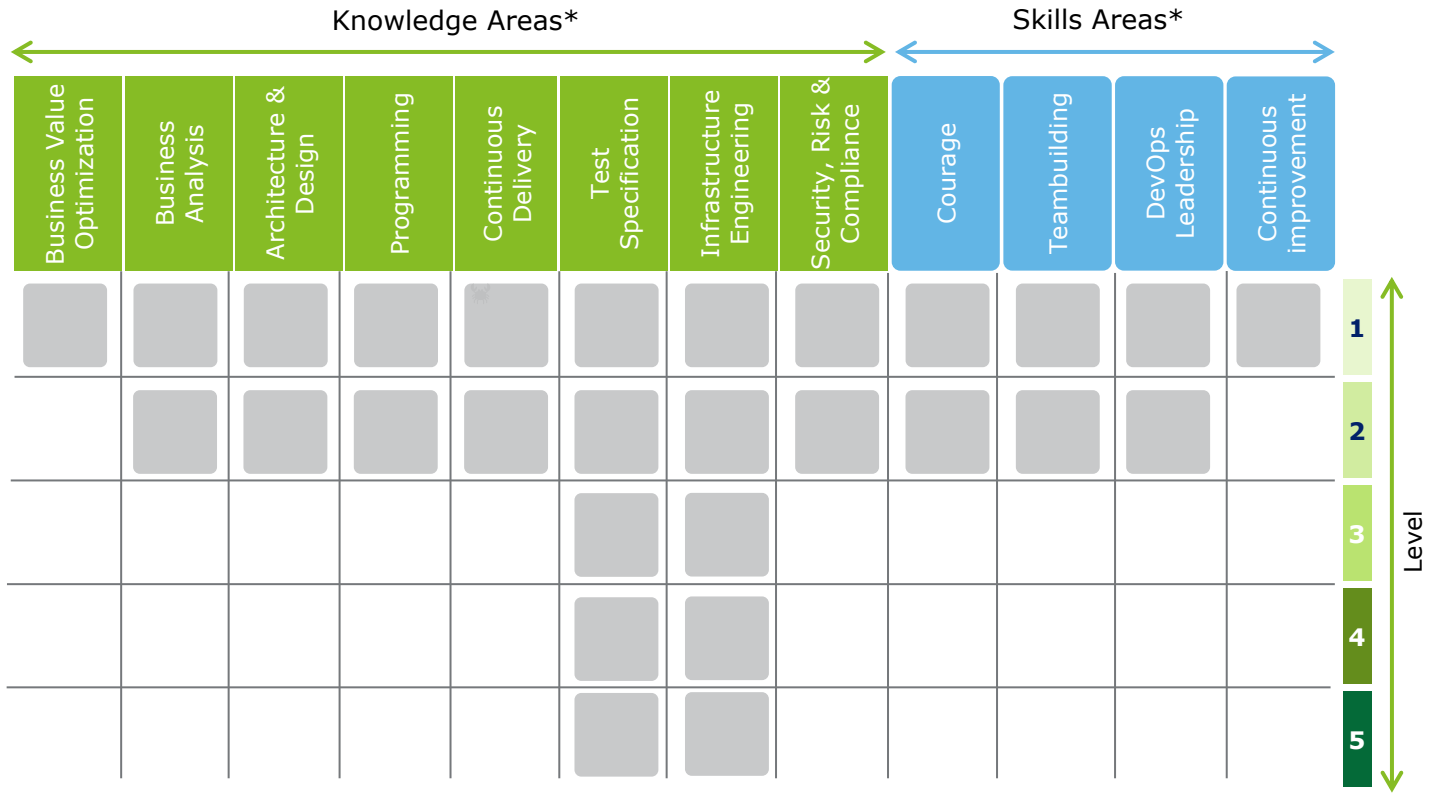
The **Scrum Master** facilitates the team on processes & approach, manages impediments and enables continuous improvement

*The scope of DevOps engineer role depends on the maturity of the shared infrastructure function.

© 2022 Deloitte The Netherlands

T-shaped profiles

Ideally, DevOps team members have a T-shaped profile, teams have a combination of different profiles covering all knowledge and skills areas



Why we advise T-shaped profiles

A T-shaped profile entails that a team member covers different knowledge areas and skills in varying levels of expertise.

A team with T-shaped profiles does not have a hierarchy since everyone's skills and knowledge complement each other.

A lack of hierarchy brings a team closer together and creates a sense of shared ownership.

Level 1 — Novice
Strict obedience to rules, no experience, little situational perception, no discretionary judgement

Level 2 — Competent
Still limited with situational perception, knows the aspect guidelines and treats all attributes and aspects separately, yet equally

Level 3 — Proficient
Sets priorities, actions are seen partly in longer term goals, deliberate planning, standardized procedures

Level 4 — Expert
Perceives deviations from the normal patterns, makes decisions more easily, assesses situations as part of the "big picture"

Level 5 — Master
Has a wealth of experience, creative solutions and visions, breaks the rules when needed, uses analytic approaches sparingly, makes good decisions quickly yet professionally

*DevOps Agile Skills Association (DASA)

© 2022 Deloitte The Netherlands

The mindset of a DevOps team member

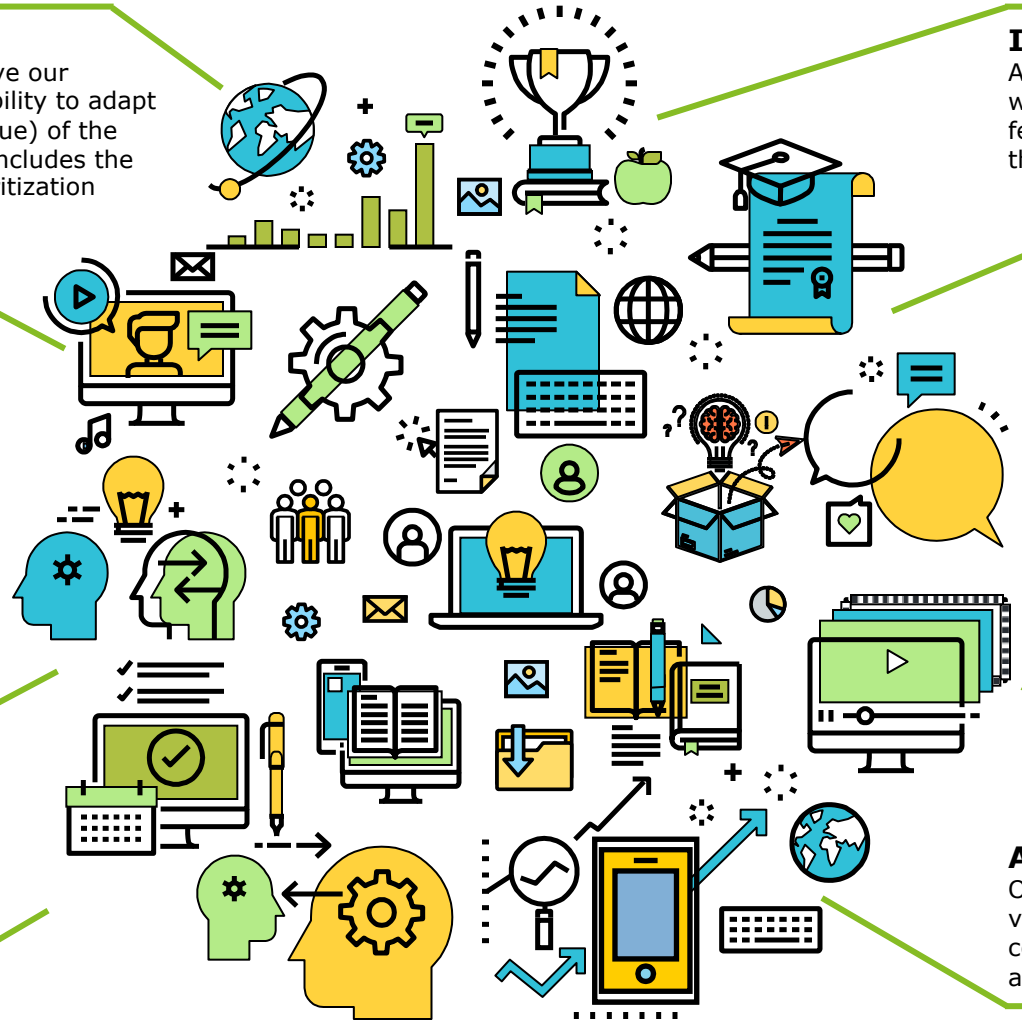
DevOps team members foster certain cultural aspects contributing to the end-to-end ownership of services

A mindset of effectiveness
We continuously improve our delivery to improve our effectiveness. We define effectiveness as our ability to adapt to "market" circumstances and the success (value) of the product features delivered. Note that this also includes the effectiveness of activities, such as backlog prioritization

A mindset of taking responsibility
All members of our team are responsible for the complete product, which includes the full delivery cycle as well as operating/providing customer support throughout the lifecycle of the product in a collaborative mindset

An engineering mindset
We have the desire to utilize our knowledge, skills, and creativity to solve problems, implement product features, and optimize our delivery process. We do not settle for the current status quo. We strive to improve our craftsmanship

Build quality in
Quality is built-in from the initiation of the teams up to discharge. It is at the heart of every activity. It is never compromised. We value full transparency



Inspirational and fun environment
An environment in which people perform at best, where they feel inspired, where they want to be, feel welcomed and are encouraged to think out of the box

Continuous learning & Continuous improvement
We have the desire to explore and learn in all activities we do. We strongly believe that working together, transparency, and sharing knowledge is vital. We care about our job enough to not pass the buck, we want to learn all the parts as a whole and not just our little world

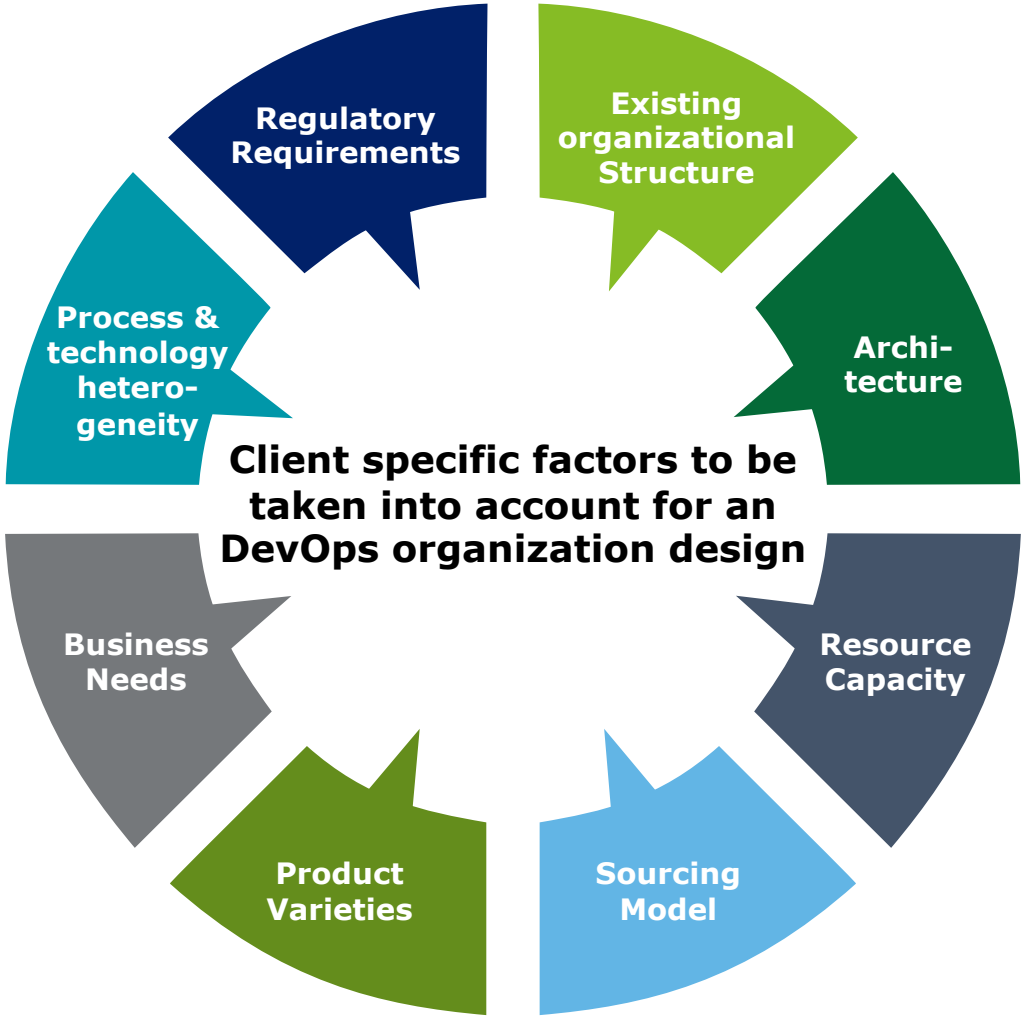
Experimentation & Risk taking
We always conduct experimentation using solid methodologies to ensure ideas are evaluated on the real value instead of the assumed value

A mindset of product thinking
Our application is our product. It must deliver value when it runs in production. We need continuous improvements to ensure the application delivers value now and in the future

© 2022 Deloitte The Netherlands

Factors influencing DevOps organization design

DevOps theory doesn’t always apply to practice, client specific factors need to be taken into account for an applicable and effective DevOps organization design



DevOps organization design is client specific

Setting up DevOps teams or an entire DevOps organization requires understanding of existing, but also future organizational structures.

Client specific factors might increase the complexity and effort that is required to transform towards a DevOps organization.

There is no “one-size-fits-all” DevOps organization design. Client specific factors must be taken into account.

Example considerations for an effective “to-be” DevOps organization design are:

- Keeping some functional hierarchy intact to facilitate collaboration with the enterprise
- Re-architecting the technology stack to enable DevOps practices
- Adhering to some degree of separation of duties to comply with regulations

© 2022 Deloitte The Netherlands

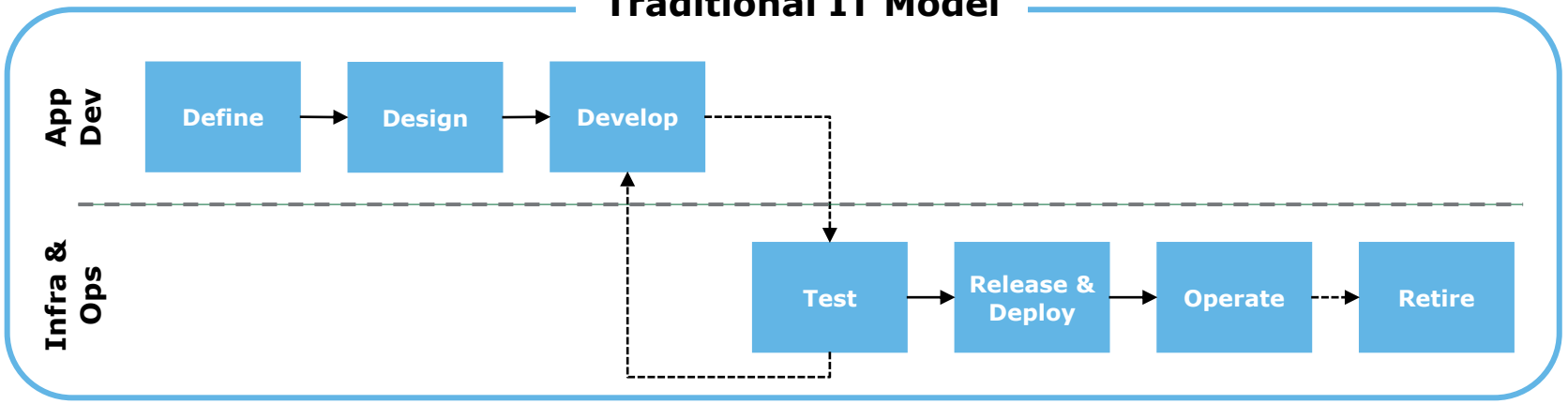
Process

© 2022 Deloitte The Netherlands

The DevOps model is significantly different from the traditional IT model

DevOps integrates the application lifecycle into an end-to-end, iterative process

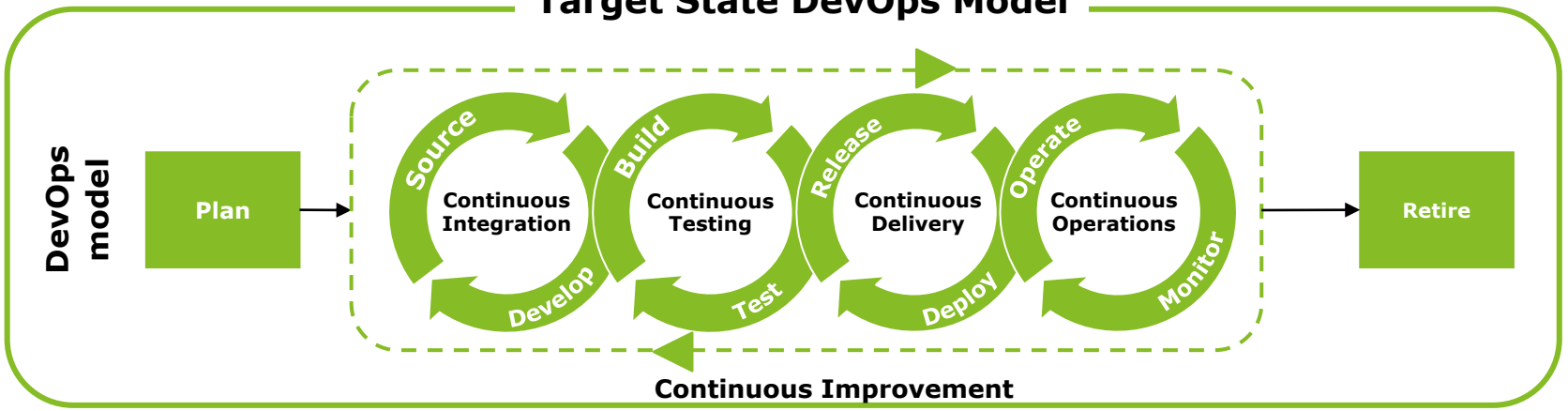
Traditional IT Model



Implications

- Straightforward sequential process assuming all is known
- Big chunks of work
- Maximizes each process-step, big-bang delivery
- Many separated functions (silos) with (manual) handoffs
- Specialization (I-shaped roles)
- Rigid change ability

Target State DevOps Model



Implications

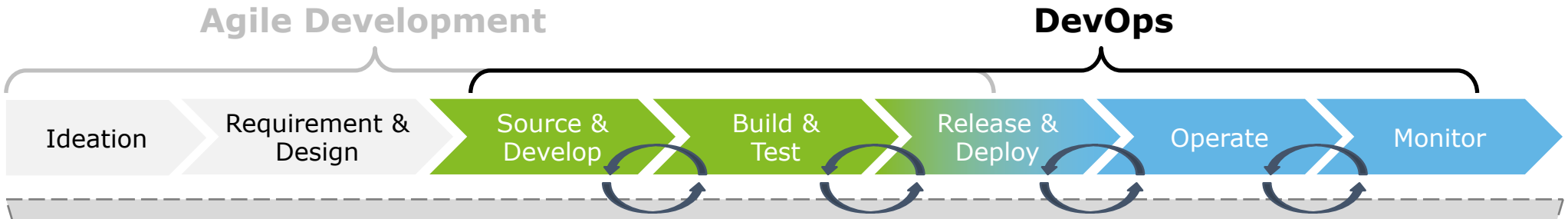
- Complex iterative process to manage unknowns
- Small chunks of work
- Maximizes flow, incremental delivery
- Fewer handoffs (less silos)
- Generalists (T-shaped roles)
- More flexible to adapt to change

© 2022 Deloitte The Netherlands

DevOps leverages technologies to automate the software delivery process

While extending agile, DevOps optimizes the software delivery process by leveraging CI/CD which automatically promotes developer's source code to operational solutions

Software Delivery Process



DevOps Practices

DevOps **extends the Agile process** and puts emphasis on the software delivery cycle and operations…

… its practices focus on **bridging the stage gate gaps** between phases to accelerate throughput by promoting more frequently with smaller products…

… next to this, DevOps practices incorporate **feedback loops** continuously in the process for value creation and learning by experience

The CI/CD Pipeline

CI/CD pipelines integrate process into technology

- **Automates almost everything -** Automation drastically reduces time, effort, and risk of human errors
- **Done means released -** Deliver releases for pre-deployment or deploy new releases to productions in minutes instead of months
- **Everything in Version Control -** Versioning ensures that no work gets overwritten and that the latest versions are built upon
- **Builds Quality Into the Process –** The quality of every deliverable is guaranteed errors and problems are detected early

© 2022 Deloitte The Netherlands

Technology

© 2022 Deloitte The Netherlands

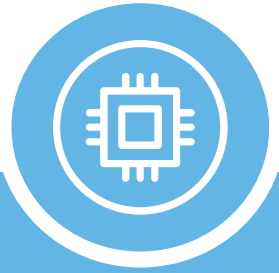
Architecture design in a DevOps implementation

Realizing the full potential of DevOps requires a flexible, secure and agile technology stack, which often brings the need for architectural redesign

*Increased **end-to-end delivery speed and maturity** poses **increased demands** on an organization's technology stack. **Re-architecting for DevOps** therefore requires decisions on some key technology areas:*

Modularity & Independent Deployment

External dependencies on other technologies and processes can put at risk the rapid deployment of new features. Minimize dependencies by designing modular systems.



Resource & Infrastructure Management

Even with the fastest deployment pipeline, going live will not be possible without the proper supporting infrastructure. Effectively managing (standardized) resources is critical for effective DevOps.

Fit-for-purpose Supporting Technology

The supporting technology to develop a CI/CD pipeline must be consistent with the organization's objectives, available capabilities and maturity level.

Responsible DevOps

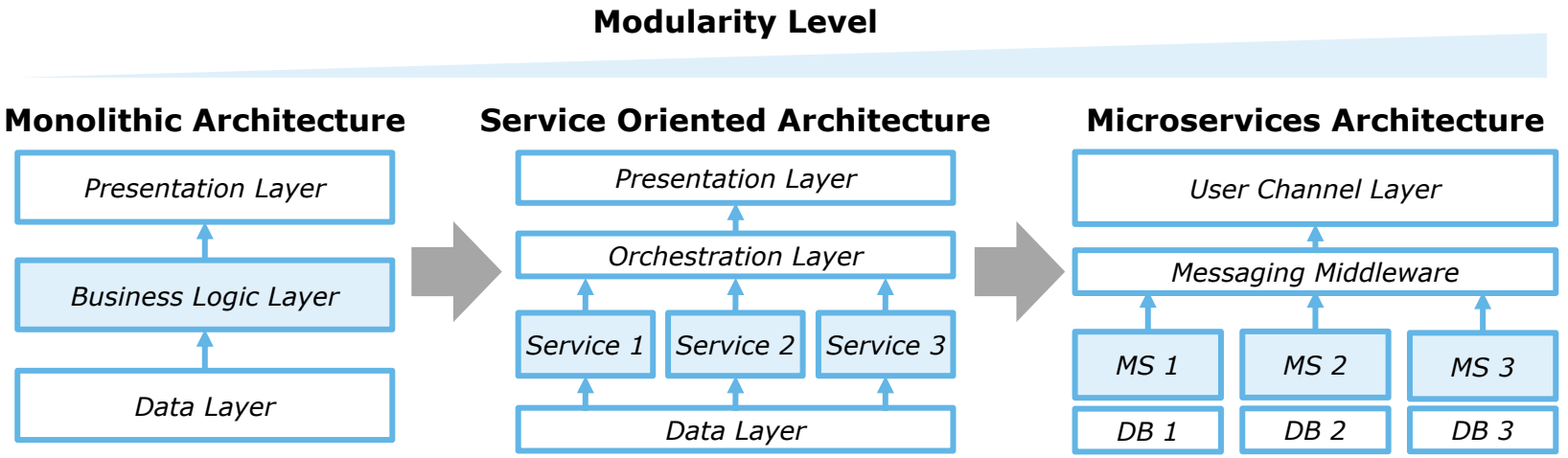
In modern IT landscapes, speed cannot be achieved at the cost of security, privacy or compliance. Integrating a risk-based approach in the DevOps process and pipeline is fundamental to ensure full control.

© 2022 Deloitte The Netherlands

Modularity as a mean to boost the impact of DevOps

By decreasing technical & functional dependencies among components, modularized systems can empower DevOps and further increase delivery speed and quality

Modularity is a **spectrum,** whose level can be influenced by decisions on both **software functional segregation** and **system architecture** (e.g. having a messaging layer, splitting DBs etc.)



Features of modular systems

Impact on DevOps

Independent Deployments

Small components enable organization to create small, autonomous teams (e.g. Amazon rule), which are **free** to **deploy** their changes **independently**, **without** having to **wait** for the **entire application** to be deployable.

Contract-based Coordination

By enforcing standard services' interfaces (i.e. how the different services talk to each other), delivery teams can **decrease** the **amount** and **impact** of **inter-team dependencies and make collaboration more efficient**.

Localized Failure

As **failures** are **localized**, teams can more confidently **deploy stand-alone software** since, if the service is faulty, the system entirety will not be compromised.

© 2022 Deloitte The Netherlands

Cloud is an accelerator for DevOps

Though DevOps can be applied in an on-premise or hybrid environment, cloud infrastructure is an accelerator that enables high levels of automation

Cloud

DevOps can be **accelerated** by leveraging resource automation principles that are widely applied in the **cloud**. Several capabilities of cloud infrastructure offer **distinct advantages** when implementing DevOps:

- Easy and rapid creation and replication of environments ensures **scalability** and **business continuity**
- Fully automated (de-)provisioning of infra using **infrastructure-as-code** enables resource flexibility, resulting in a **resilient infrastructure**
- Services provided by cloud vendors focus on **high usability** and **easy integration**, reducing the additional expertise required in teams

Ease of scaling	
Environment flexibility	
Impact on people	

On-premise

Achieving an **agile** and **modular** technology stack is definitely possible in an on-premise environment. However, there are **some disadvantages** to building and managing your own infrastructure:

- **Limited control** over networking and security policies often prevent easy replication of environments and requires alignment with a wide range of stakeholders
- Hard boundaries on resource flexibility due to **hardware constraints** and integration with **legacy systems** limit an infrastructure's resiliency
- On-premise often leads to a significant **increase in scope** of DevOps teams, and require experts with **specific knowledge**

© 2022 Deloitte The Netherlands

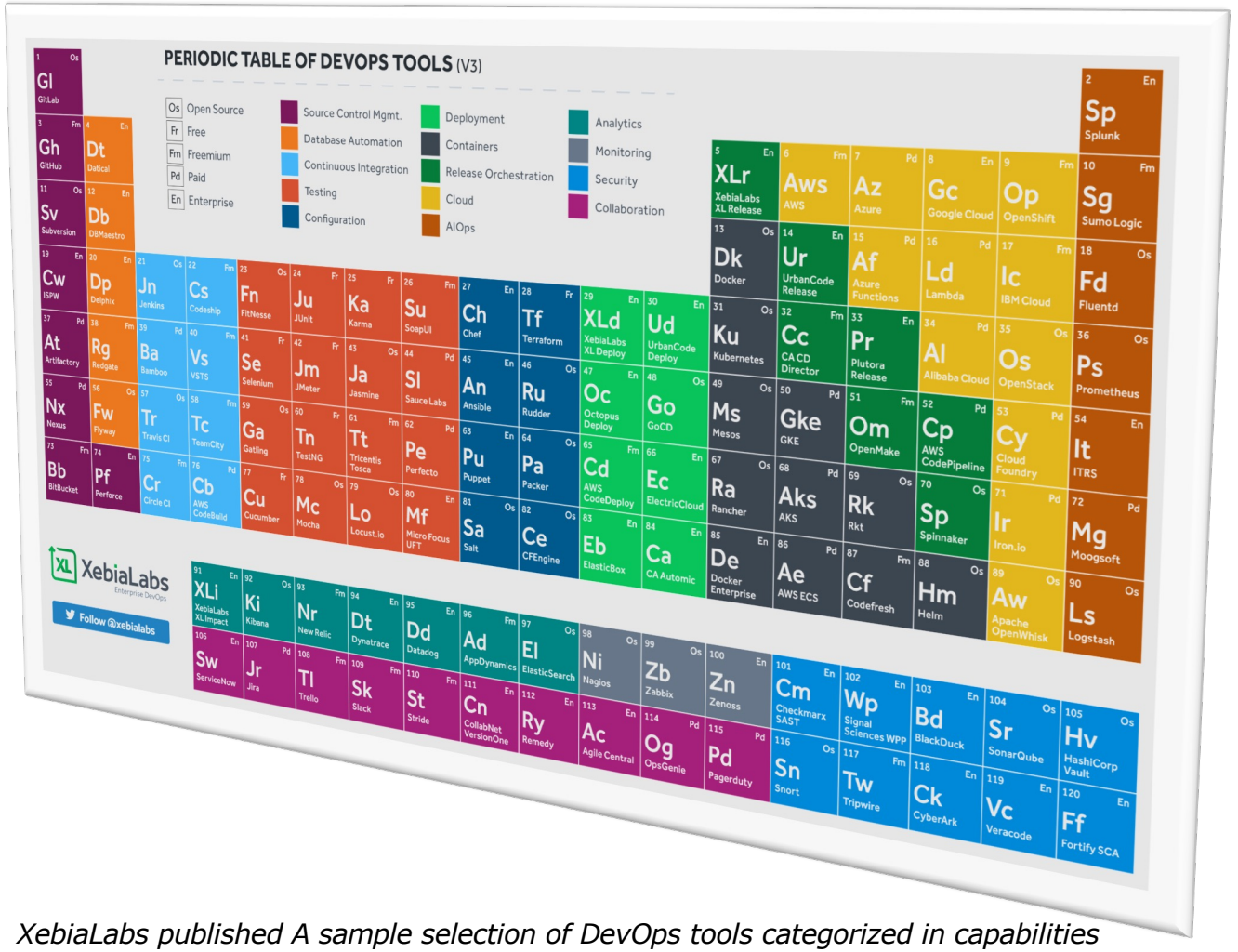
A legion of tools are available to support DevOps practices

As DevOps is a tool intensive practice, a thorough tool selection incorporating client maturity is a crucial part of the DevOps transformation

DevOps is a tool intensive practice

The voluminous amount of tools available, delivering one or multiple capabilities brings consequences when transforming towards a DevOps organization

- Selecting the right tools requires an iterative approach (a procedure per capability)
- Selection is, among others, based on engineering skills, prior experience, or tools (architecture) already in place



XebiaLabs published A sample selection of DevOps tools categorized in capabilities

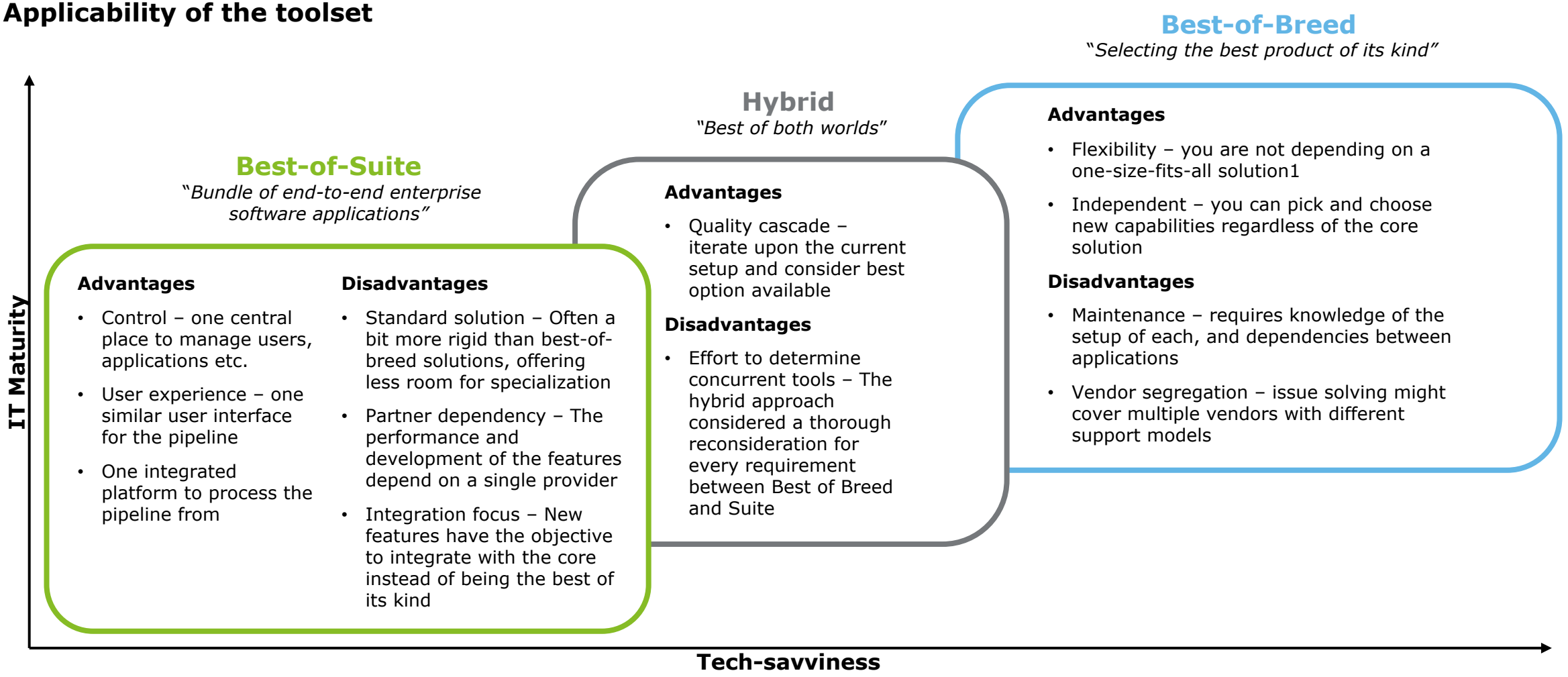
Source: https://xebialabs.com/periodic-table-of-devops-tools/

© 2022 Deloitte The Netherlands

Patterns to setup a CI/CD pipeline

Selecting the right set of tools (Best-of-Suite, Best-of-Breed or hybrid) for the CI/CD pipeline depends heavily on IT maturity and tech-savviness of the organization

Applicability of the toolset



© 2022 Deloitte The Netherlands

A basic functional flow through a CI/CD pipeline

A CI/CD Pipeline can be build in various ways considering the desired tooling patterns, covering the same functional flow with different tools and integrations between them

Source & Develop

Distributed source code repository and version control system manages code changes during software development

Source Code Repository and Version Control

Agile Planning and Collaboration

Collaboration environment supports the delivery process and planning of the DevOps Teams

Build & Test

Build and test automation process the application code based on the latest changes in the source code repository

Code Build and Automated Testing

Test Strategy, Execution and Reporting

Test repository manages test strategies, test cases, test execution and reporting of test results

Release & Deploy

Central repository manages and versions the released application artifacts and dependencies

Artifact Repository

Automated Deployment

Deployment automation deploys the released application artifacts and dependencies to target environments

Operate & Monitor

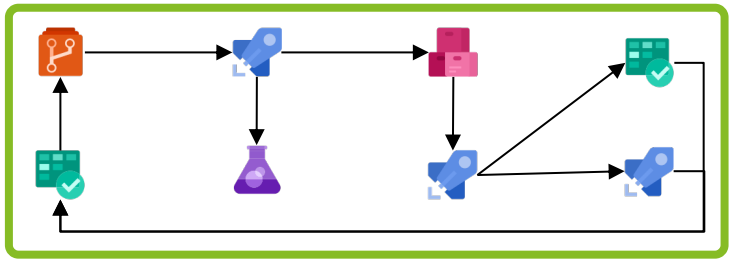
Monitoring the performance of the team in the software delivery process using DevOps metrics and KPIs

Software Delivery Process

Software Operations

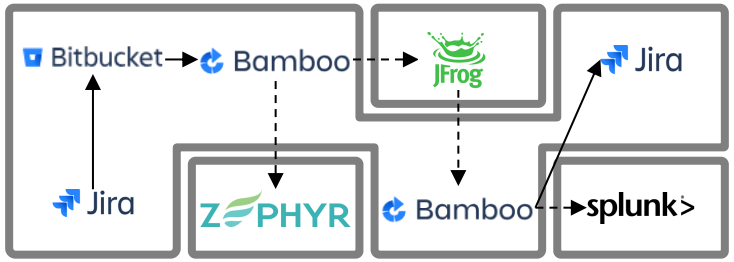
Monitoring of the deployed system's health & performance using application and infrastructure monitors

Best-of-Suite



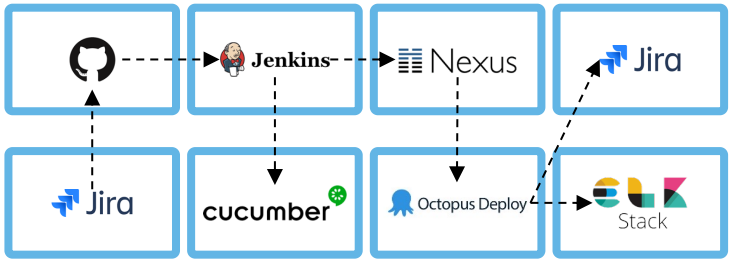
Azure DevOps covers the full extent of the CI/CD pipeline, with no external integration required

Hybrid



Only a few interfaces are required as Atlassian's suite covers the majority of required functionality

Best-of-Breed



The pipeline orchestrator (Jenkins) becomes the central component to integrate all applications

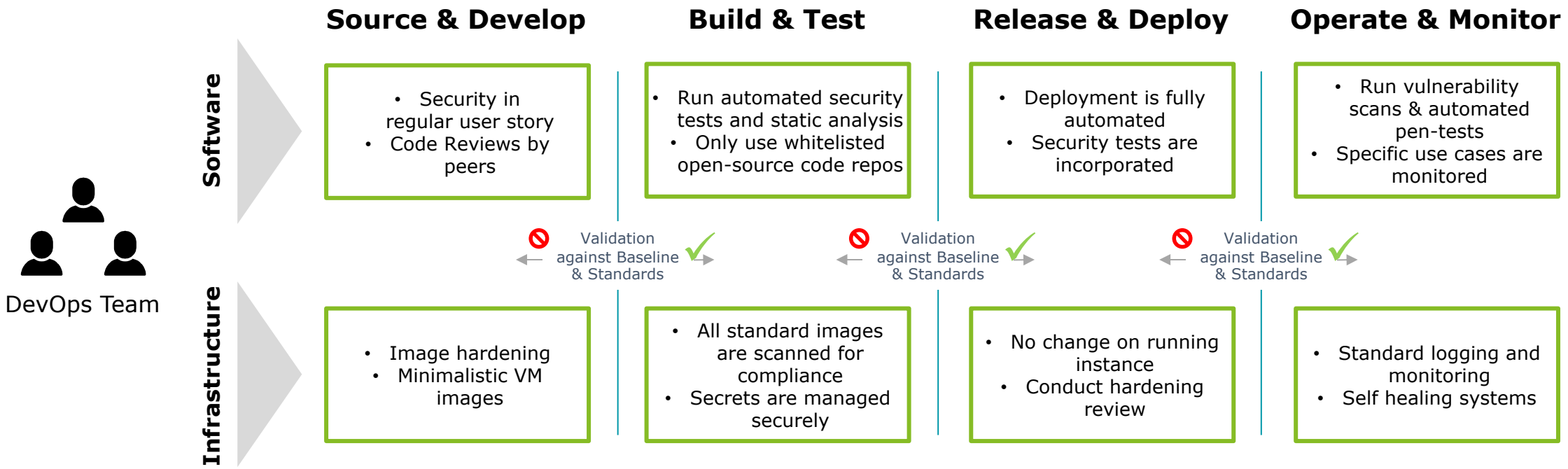
Same suite, no interface

Interface between tools/suites

© 2022 Deloitte The Netherlands

Security at every step of the software delivery lifecycle

DevOps requires a strong focus on integrating security automation in the deployment pipeline, as well as including security* in the regular development process



The items above are examples of DevOps activities/controls that can be put in place across the different stages of a CI/CD pipeline, for both software and infrastructure

** Although security is an inherent component of every step of the DevOps cycle, security begins by incorporating it into the design of the application.*

© 2022 Deloitte The Netherlands

Operating Model

© 2022 Deloitte The Netherlands

The changing IT Operating Model

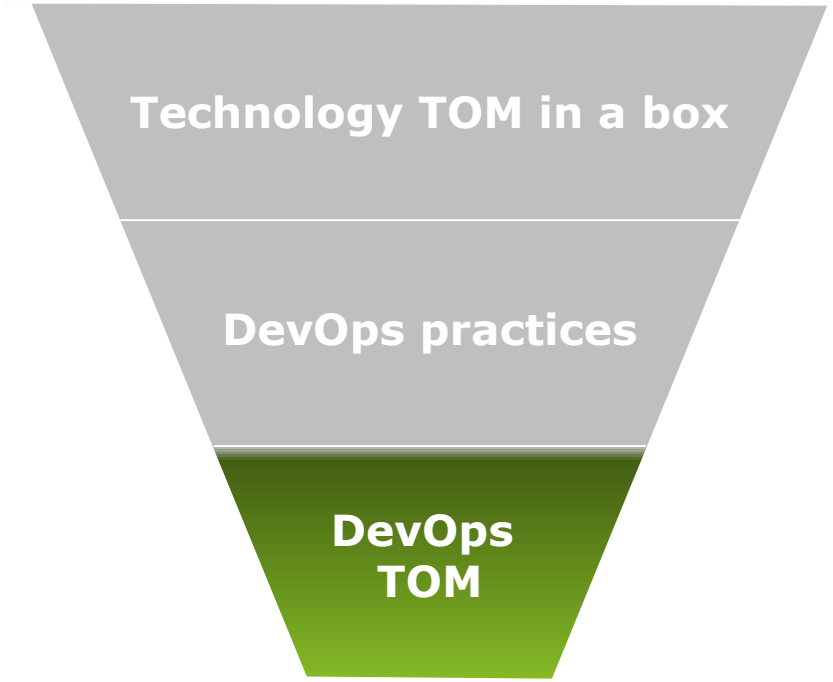
DevOps disrupts multiple dimensions of the IT Operating Model; synergies arise when combined with next generation technologies and ways-of-working

	Organizational Model	Architecture Model	Delivery Model	Release Model	Practice Model	Hosting Model	Sourcing Model	Support Model
Traditional	**Hierarchical Organization**	**Monolithic**	**Waterfall**	**Annually & Quarterly**	**Process Driven**	**Physical Datacenter**	**In-house**	**Tier Based** (L1 → L2 → L3)
Evolved	**Product Line Organization**	**N-Tier**	**Agile**	**Monthly**	**Practice Driven**	**Virtual Machines**	**Traditional Outsourcing (Towers)**	**Skilled Support**
Next Gen	**Autonomous Teams**	**Microservices & API**	**Continuous Delivery**	**On Demand**	**Value Stream**	**Cloud & Containers**	**Service Brokering**	**Self Service & Self Help**

© 2022 Deloitte The Netherlands

DevOps Target Operating Model (TOM)

A DevOps TOM lands the DevOps way-of-working in the organization by describing the DevOps practices using Deloitte's framework 'Technology TOM in a box'



The **Technology TOM in a box** is a holistic framework to describe the governance structure of an IT organization and how it functions as an entity

The capabilities of the **DevOps practices** will be described through the dimensions of the Technology TOM in a box

The **DevOps Target Operating Model (TOM)** outlines the governance structures on how an IT organization should govern and operate the DevOps practices

© 2022 Deloitte The Netherlands

Defining a DevOps Operating Model

There is a logical order to defining and documenting the interrelated dimensions of a DevOps Target Operating Model

The DevOps TOM is a number of interrelated dimensions that describe the governance structure of an IT organization and how it functions as an entity.

1

What *is the overarching mission, which services are the DevOps organization going to deliver to achieve this mission, and which personas produce and consume these services?*

- Mission
- Personas
- Services

2

How *do capabilities deliver these services, and, on a* ***high-level****, how are these capabilities structured, sourced and funded by the DevOps organization?*

- Capabilities & Skills
- Organization Structure
- Governance & Decision Rights
- Sourcing & Ecosystems
- Funding & Charging

3

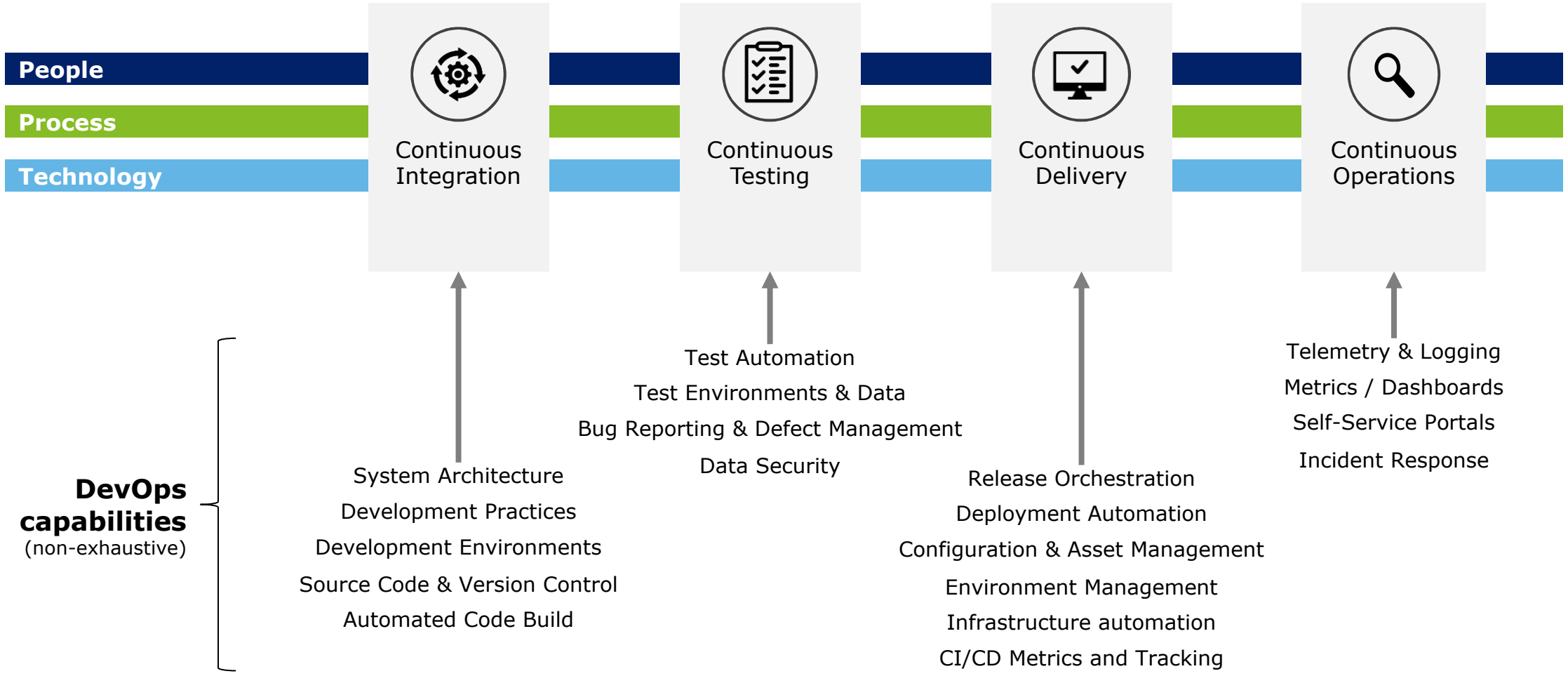
How *do DevOps teams collaborate on a* ***detailed level*** *to deliver these services; individual roles & responsibilities, processes, tooling and metrics?*

- Roles & Responsibilities
- Tooling & Automation
- Culture / work style
- Processes & Interactions
- Metrics & KPI's
- Collaboration & Location

© 2022 Deloitte The Netherlands

Detailing the DevOps practices

The DevOps TOM describes capabilities that are part of the DevOps practices



© 2022 Deloitte The Netherlands

DevOps operating model differentiators

The DevOps way-of-working has several key changes on the IT operating model that differentiate from the traditional way-of-working

Differentiators of the DevOps TOM

People

Impact:

- **Service Oriented Organization**
  Resources organized around services, focused on value
- **Service Brokering**
  As-a-service delivery of composed products with an ecosystem of partners
- **Autonomous teams**
  Self-organizing teams, autonomously in their decision making
- **Cross functional teams**
  Multiple disciplines in a team that share E2E responsibility

Process

Impact:

- **Release On Demand**
  DevOps practices (CI/CD) for iterative and frequent releases
- **Continuous delivery**
  Slicing the work into smaller chunks that add value to customers
- **E2E traceability**
  Greater visibility and transparency on service delivery
- **Value stream**
  Siloes broken down between business and IT
- **Self Service & Self Help**
  service request portals automate request fulfillment

Technology

Impact:

- **Automate everything**
  Tools automate processes, minimize handoffs and maximize checkpoints
- **Modular App. Architectures**
  Architecture (e.g. Micro-Services & API) enables autonomous teams to develop and release independently
- **Infrastructure as code**
  Next gen hosting (e.g. Cloud & Containers) enables automated environment provisioning
- **Venture-Capital Budgeting**
  Funding dependent on minimal viable product, and its performance

Source: Deloitte Technology TOM in a box

© 2022 Deloitte The Netherlands

How do I implement DevOps?

© 2022 Deloitte The Netherlands

DevOps transformation approach

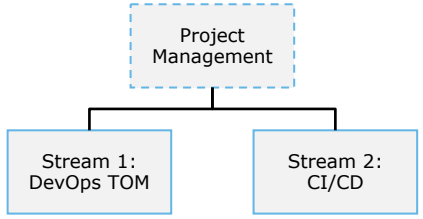
The transformation approach outlines the Deloitte service used to collaboratively drive and coordinate a successful DevOps transformation, consisting of four different phases

1 Start Project
3 – 6 months

Start with two project tracks; design and execute

Key objectives

- Define mission and IT services in scope
- Start design track for DevOps TOM and roadmap
- Start execution track to for CI/CD implementation

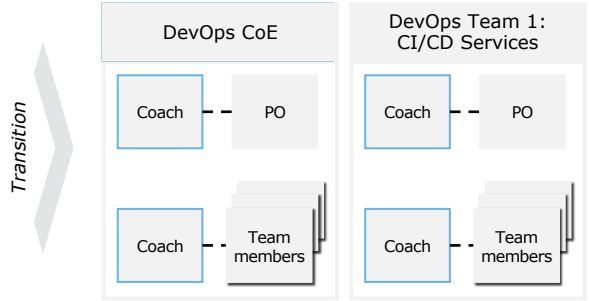


2 Transition Delivery
3 months

Transition into a bi-modal organization where coaching and co-creation is key

Key objectives

- Setup a DevOps CoE as driver of the transformation
- Onboard CI/CD Services as the first DevOps team
- Replace traditional governance with DevOps ceremonies

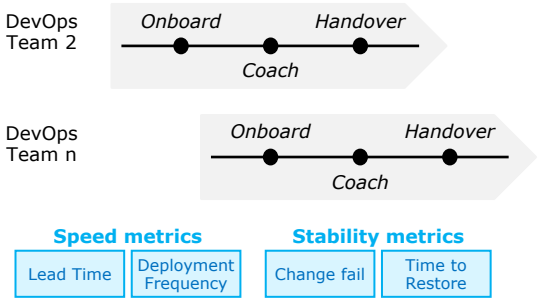


3 Coach & Scale
*6 – x months**

Onboard, coach and handover new virtual teams to scale the bi-modal organization

Key objectives

- Shape the DevOps TOM by learning from experiments
- Experiment with CI/CD to accelerate value delivery
- Show value using DevOps Metrics

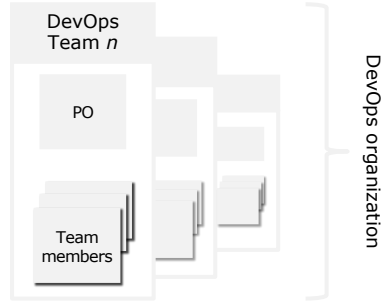


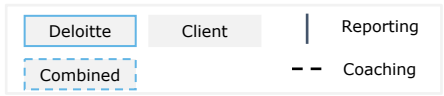
4 Roll-off
1 – 2 months

Roll-off once the client is mature enough to continue themselves

Key objectives

- Stop coaching efforts once the organization is mature enough
- Handover of transformation responsibilities





© 2022 Deloitte The Netherlands

* *Depending on # of teams to onboard*

What are the challenges of implementing DevOps?

DevOps majorly challenges skills of everyone involved: management and team. It may lead to development slowdown and will not compensate for lack of responsibility

People

Challenges:

- **Skill Challenge:** Difficult to replicate skills across every DevOps team in your organization and acquiring people with the DevOps skills may be challenging
- **Management Challenge:** DevOps can mean management challenges for your team leads
- **No Magic:** DevOps will not compensate for potential lack of responsibility in your organization

Process

Challenges:

- **Lack of Overview:** Progress and stability are spread across the teams and overview may lack
- **Dev Slowdown:** Operations may hamper progress in development
- **Self-organization Challenge:** Clear Service-Level Structures in operations may be challenging

Technology

Challenges:

- **Application architecture:** Transforming an organization from being monolithic to work with microservices and API's
- **Choice of Tools:** difficult to choose appropriate tooling when knowledge about the technology behind it is lacking
- **Integration with existing IT landscape:** Pipeline platform integration with enterprise IT (network, infra, hardware, security, applications) is often experienced as difficult

© 2022 Deloitte The Netherlands

Key Lessons Learned

During our engagements we gathered the following key takeaways that we will bring to future projects

People	**You cannot "buy DevOps"**	**Management support is crucial**	**Break down silos**	**Assign champions from client**
	DevOps adoption cannot be bought and "bolted on" the existing organization. It requires a cultural shift around how people deliver their work	Management involvement is crucial in the DevOps transformation, as change starts and stops with them	Break down silos. Not only between departments, but also between organizations	Ensuring the support of the client can be accelerated by having a champion from their side spreading the DevOps culture and principles
Process	**Consider secondary impacts**	**Collaboration is key**	**DevOps ≠ Agile**	**Focus on E2E responsibility**
	Product roadmaps will be impacted and delivery bottlenecks reduced. New budget to build a DevOps organization will be needed	DevOps requires close collaboration across dev, test, operations and business teams to effectively deliver value to the organization	DevOps can be seen as an extension of Agile, with the same level of agility driven into development, test and operations	Limit handovers as much as possible, teams must adopt an end-to-end responsibility for the product or service they deliver
Technology	**Modern architecture is critical**	**Show value quickly**	**DevOps ≠ Automation**	**Use Cloud as an accelerator**
	Platforms built on modern architectures based on modular design, decoupling and good componentization enables deeper adoption of DevOps	'Prove' the DevOps concept by demonstrating working solutions early and often (e.g. CI/CD tooling)	Release and Deployment Automation or App Release Automation are only a part of DevOps. End-to-end automation is key	Ensure parity between cloud and on-premise implementations (e.g. Azure DevOps)
Operating Model	**Collaborate with Tech Stream**	**Change incrementally**	**DevOps ≠ Organization**	**DevOps journey is client specific**
	Design the DevOps operating model in parallel and close collaboration with the technology implementation	Apply an agile approach for adopting DevOps and introduce change incrementally with focus on the outcomes	There are different organizational patterns to setting up DevOps and it doesn't always have to be making it a separate organization	DevOps target operating model transformation depends heavily on where the client is in their DevOps journey

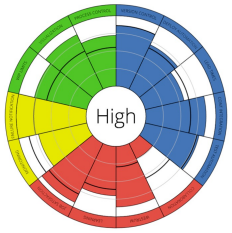
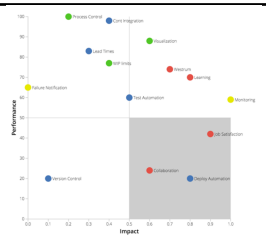
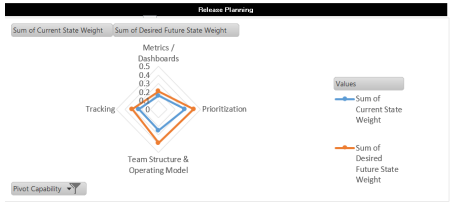
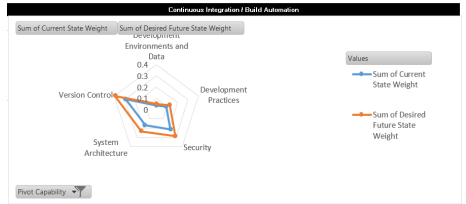
© 2022 Deloitte The Netherlands

Deloitte Accelerators

© 2022 Deloitte The Netherlands

Deloitte DevOps Maturity Assessment Offerings

To understand the current state DevOps capabilities and to identify areas for improvement, we have two assessment methodologies available.

	DORA	Deloitte DevOps Maturity Assessment
What is it?	DORA provides a SaaS questionnaire that benchmarks DevOps performance against 2000+ leading Enterprises across industries	Deloitte's DevOps Maturity Assessment (DDMA) is an extensive questionnaire for assessing current state against desired future state maturity of DevOps capabilities across the DevOps domains: from Release Planning to Continuous Deployment and Monitoring.
Benefits	▪ "Gold standard" for DevOps assessments ▪ Compare your results against others in industry ▪ Two assessments included – one to baseline and one to measure progress ▪ Provides priorities for capability improvement	▪ 180 questions along each of the DevOps domains (Release Planning, Continuous Development, Continuous Integration, Continuous Testing, Continuous Deployment, Continuous Monitoring) ▪ Assesses maturity against desired future state ▪ Identifies areas for capability improvement
Cost	Deloitte receives a 30% discount from DORA; will be an additional cost on top of pilots	Included in price of DevOps KickStart or DevOps Dojo
Sample Output	 **Industry Benchmark**  **Capability Prioritization**	 **Release Planning Maturity**  **Continuous Integration Maturity**

© 2022 Deloitte The Netherlands

Deloitte Global Accelerators

Our Global DevOps Community of Practice has a wide variety of accelerators available that we can use in our engagements

Learning

Learning Series

Basic introduction course to various DevOps practices

Learning Resources

A collection of documents to assist learning DevOps and specific elements or specific vendors

Videos & Demos

A collection of videos and demos regarding Deloitte methodologies and instructions for DevOps tooling

Sales Materials

Proposal templates & DevOps brochures

Templates & brochure to help you kickstart your DevOps proposal

DevOps Qualifications

'Quals' to help you display Deloitte's capability to deliver DevOps transformations, including tooling

DevOps Case studies

Case studies of client engagements, with success stories and demos. The Client demo can showcase DevOps automation capabilities

Tools and Enablers

DevOps Local toolkit

An integrated toolkit of local DevOps tools to gain hands-on experiences

Deloitte supplied tools

Tools that can be supplied for client engagements:

- Agile Manager
- HP Application Lifecycle Management
- Fortify
- JRebel
- Performance Center
- SonarQube
- Unified Functional Testing and UFT Pro

Enablers

Enabling materials for specific vendors, industries, such as the Cloud Compass, PoC for SAP or Google Cloud enablers, Cards for Agility, Technology TOM in a Box

Eminence and Point of Views

Eminence

Examples of Deloitte DevOps materials published in popular media

DevOps Point of Views (PoV)

- Cloud platforms
- Collaboration tools
- Development suite tools
- Software Build tools
- Software Deployment tools
- Container persistence
- DevSecOps

Note: non-exhaustive, Global examples, which may be updated continuously

© 2022 Deloitte The Netherlands

Client example 1

DevOps journey and CI/CD pipeline implementation

© 2022 Deloitte The Netherlands

Client example 1: Global parcel delivery services company

We took our client on a DevOps transformation journey across all five dimensions to streamline the software delivery lifecycle of their mission critical system

Process

Develop a chain of full end-to-end processes to facilitate the DevOps way-of-working and continuous software delivery

- Described CI/CD processes to operate the pipeline through all DevOps practices
- Implemented continuous feedback loops into process flows to facilitate continuous improvement
- Defined and implemented auxiliary processes to support and smoothen the execution of the DevOps lifecycle

Technology

From requirements, to tool selection, architecture design and full implementation; we built a CI/CD pipeline base on MS Azure DevOps to enable continuous integration and continuous delivery

- Automated as much as possible, while maintaining stage gates for deploying to mission critical environments
- Supported persistent configuration management to deliver tailored software to distributed, distinct production systems

Organization & Culture

Designed and implemented a fully fledged DevOps organization, along with teams covering the full lifecycle of services, as well as defining a culture to facilitate collaboration, knowledge sharing and continuous improvement

- T-shaped role descriptions for team members
- Described ways-of-working to enhance visibility and feedback
- Defined a culture based on CALMR principles, with accompanying metrics to enhance adoption

Data

Obtain as much insight, by logging all data and monitoring relevant metrics, by employing DataOps

- Gather data from CI/CD process
- Infrastructure logging & monitoring of develop, test and production environments
- Operational data logging, monitoring and analytics on operational process execution
- Provide dashboards to view and report on performance

© 2022 Deloitte The Netherlands

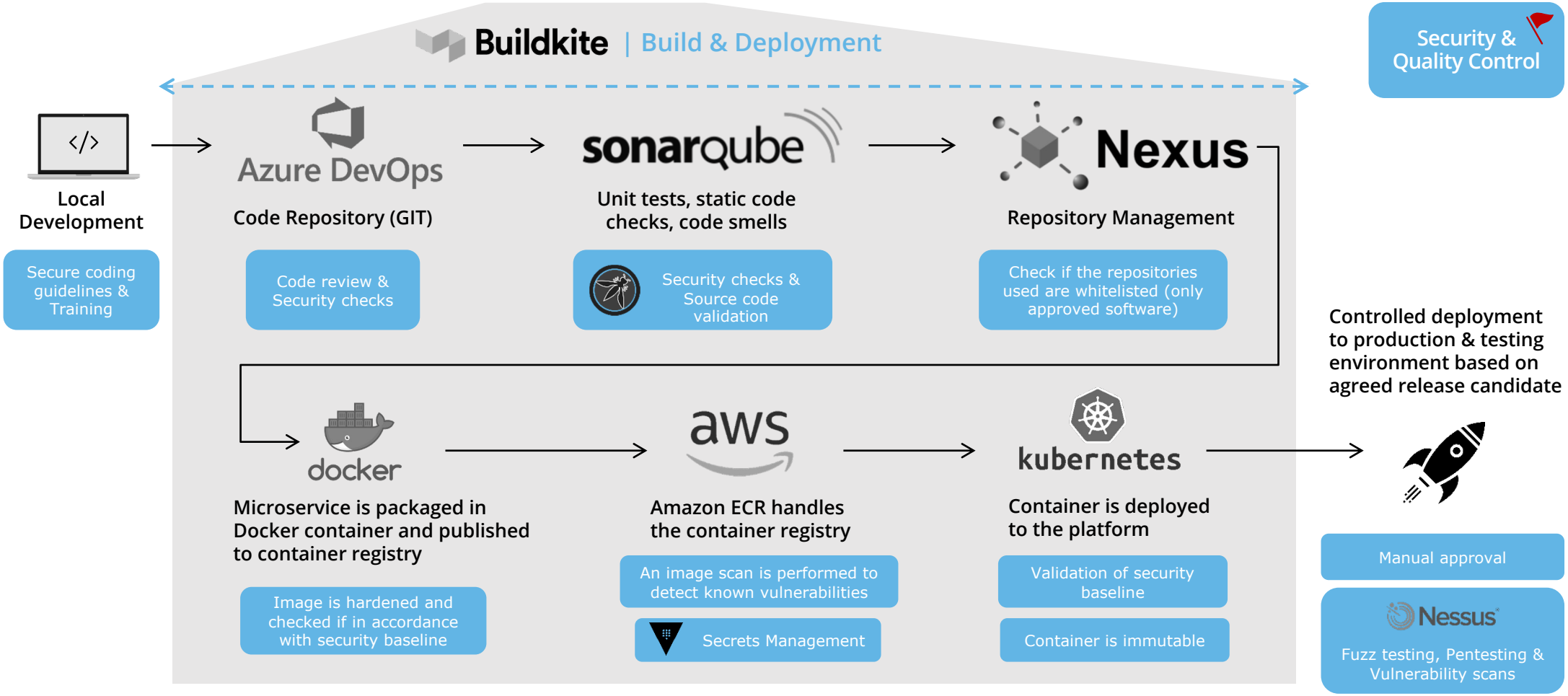
Client example 2

A CI/CD Pipeline for a Banking Platform

© 2022 Deloitte The Netherlands

Client example 2: CI/CD pipeline

Deloitte developed an Open Banking Platform as a global asset with an CI/CD pipeline to ensures continuous integration and deployment



© 2022 Deloitte The Netherlands

Client Example 3

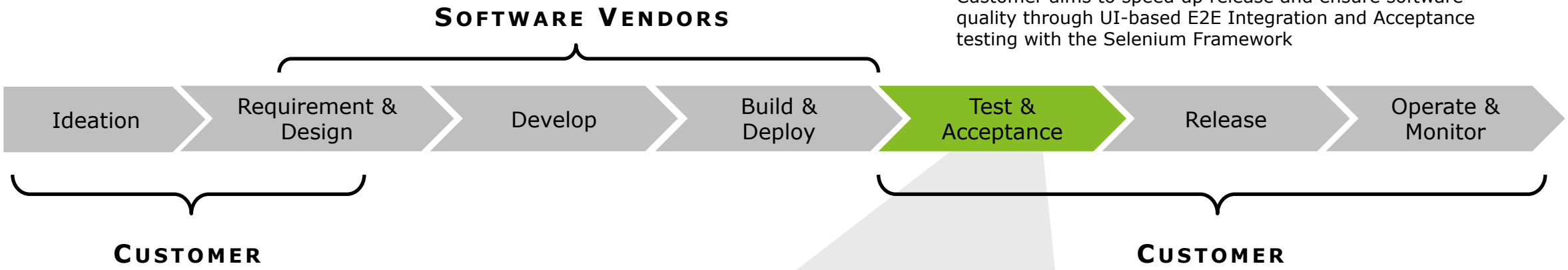
Test Automation for a Banking Application

© 2022 Deloitte The Netherlands

Client example 3: Automating End-to-End Testing

We helped our customer to setup their testing capability, a vital but time-consuming part of the software delivery process

Customer aims to speed up release and ensure software quality through UI-based E2E Integration and Acceptance testing with the Selenium Framework



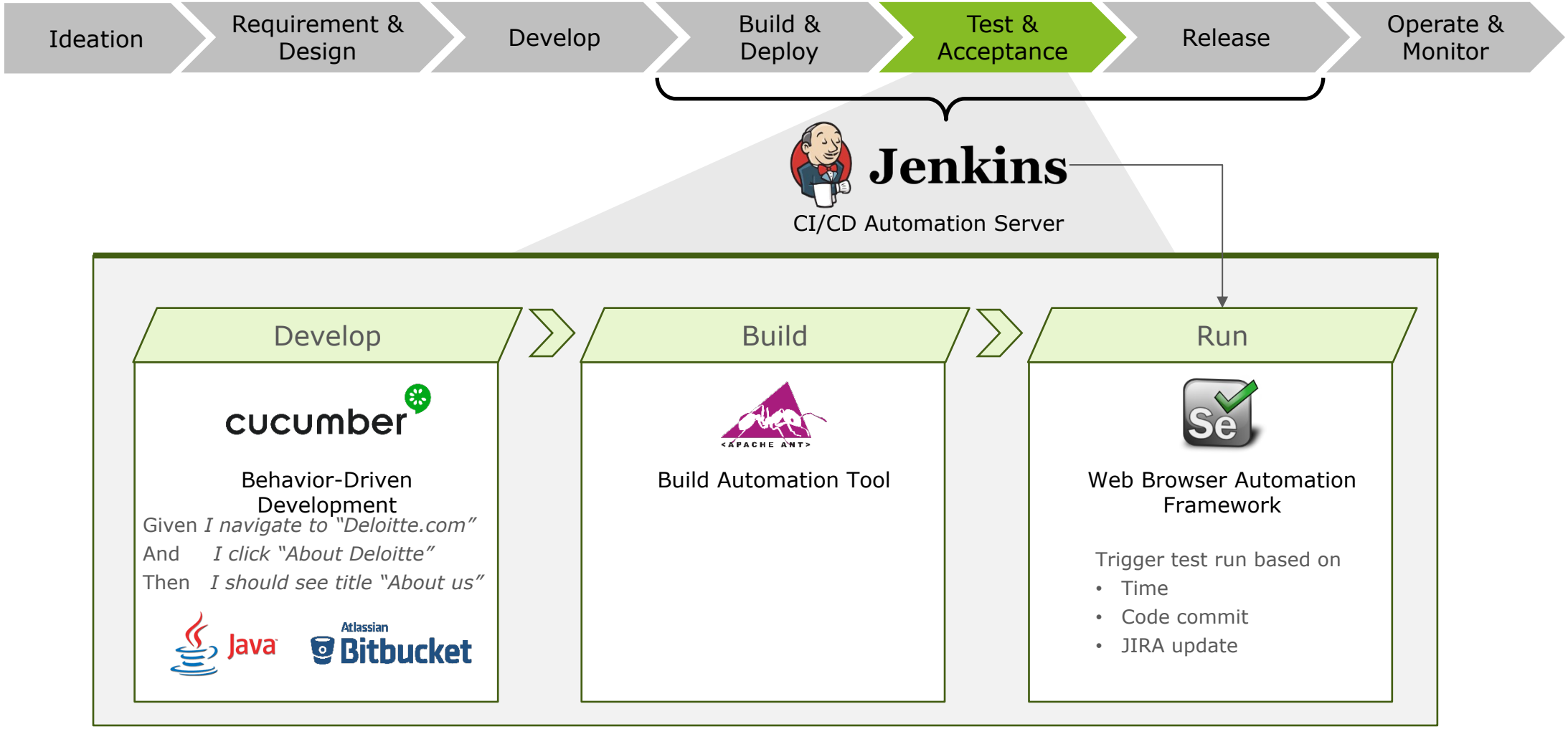
Test Phase	System	Integration	E2E	Acceptance	PAT

Type of tests	• System tests
	  • Bug testing
	  • Unit testing
	• Deployment tests
		• Deployment tests
		• Integration tests based on user stories
		• Regression tests (E2E)
				• Deployment tests
				• Business Acceptance Regression tests (E2E)
					• Non-functional tests
					  • Performance
					  • Security
					  • Disaster Recovery
					• Prepare for go-live execution

© 2022 Deloitte The Netherlands

Client example 3: Automating End-to-End Testing

Using a test automation framework based on Selenium, Deloitte delivered a total of 15 automated end-to-end test cases



© 2022 Deloitte The Netherlands

Recommended resources

In case you got excited and would like to learn more…

Books:

- **The DevOps Handbook:** How to Create World-Class Agility, Reliability, and Security in Technology Organizations
  - by Gene Kim, John Willis, Patrick Debois, Jez Humble
- **The Phoenix Project**
  - by Gene Kim, Kevin Behr, George Spafford
- **Continuous Delivery:** Reliable Software Releases through Build, Test, and Deployment Automation
  - by Jez Humble and David Farley
- **Accelerate:** Building and Scaling High Performing Technology Organizations
  - by Nicole Forsgren, Jez Humble, Gene Kim

Websites:

- https://notafactoryanymore.com/

Video's:

- John Smart (Deloitte colleague) at the DevOps Summit: https://www.youtube.com/watch?v=-Rq-fuiKNCU

© 2022 Deloitte The Netherlands

Appendices

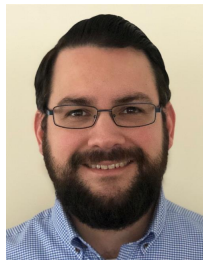
© 2022 Deloitte The Netherlands

Additional contributors

A big thanks to all our colleagues that contributed to this Point of View

Andries van Dijk
Director
Technology, Strategy & Transformation
Deloitte Consulting
+31 6 52 04 87 05
anvandijk@deloitte.nl

Tessa van den Berg
Senior Manager
Human Capital
Deloitte Consulting
+31 6 10 04 25 78
tesvandenberg@deloitte.nl



Andreas Boon
Manager
Technology, Strategy & Transformation
Deloitte Consulting
+31 6 83 33 96 09
aboon@deloitte.nl

Boris Smits
Consultant
Technology, Strategy & Transformation
Deloitte Consulting
+31 6 82 67 44 36
bsmits@deloitte.nl

Tom Pennings
Consultant
Technology, Strategy & Transformation
Deloitte Consulting
+31 6 22 17 87 17
tpennings@deloitte.nl

Jan Prüst
Consultant
Technology, Strategy & Transformation
Deloitte Consulting
+31 6 82 51 86 04
jprust@deloitte.nl

Audrey Sie
Analyst
Technology, Strategy & Transformation
Deloitte Consulting
+31 6 50 06 26 13
ausie@deloitte.nl

Jasper Lelijveld
Analyst
Technology, Strategy & Transformation
Deloitte Consulting
+31 6 50 06 25 77
jalelijveld@deloitte.nl

© 2022 Deloitte The Netherlands

Deloitte.

Deloitte refers to one or more of Deloitte Touche Tohmatsu Limited, a UK private company limited by guarantee (“DTTL”), its network of member firms, and their related entities. DTTL and each of its member firms are legally separate and independent entities. DTTL (also referred to as “Deloitte Global”) does not provide services to clients. Please see www.deloitte.nl/about to learn more about our global network of member firms.

Deloitte provides audit, consulting, financial advisory, risk advisory, tax and related services to public and private clients spanning multiple industries. Deloitte serves four out of five Fortune Global 500® companies through a globally connected network of member firms in more than 150 countries and territories bringing world-class capabilities, insights, and high-quality service to address clients’ most complex business challenges. To learn more about how Deloitte’s approximately 245,000 professionals make an impact that matters, please connect with us on Facebook, LinkedIn, or Twitter.

This communication contains general information only, and none of Deloitte Touche Tohmatsu Limited, its member firms, or their related entities (collectively, the “Deloitte Network”) is, by means of this communication, rendering professional advice or services. Before making any decision or taking any action that may affect your finances or your business, you should consult a qualified professional adviser. No entity in the Deloitte Network shall be responsible for any loss whatsoever sustained by any person who relies on this communication.

© 2022 Deloitte The Netherlands