Deloitte.

DevOps Point of View

An Enterprise Architecture perspective

Amsterdam, 2020

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 an 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 service line 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 and proposals & 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

DevOps practitioners

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

Eric Onderdelinden
Director
Enterprise Architecture
Deloitte Consulting
+31 6 83 33 98 16
eonderdelinden@deloitte.nl

Mark Maijs
Senior Manager
Enterprise Architecture
Deloitte Consulting
+31 6 83 89 02 14
mmaijs@deloitte.nl

Donald Pondman
Manager
Enterprise Architecture
Deloitte Consulting
+31 6 83 33 02 56
dpondman@deloitte.nl

Balázs Nagy
Senior Consultant
Enterprise Architecture
Deloitte Consulting
+31 6 13 31 37 65
baNagy@deloitte.nl

Boris Smits
Consultant
Enterprise Architecture
Deloitte Consulting
+31 6 82 67 44 36
bsmits@deloitte.nl

Marlies Quekel
Analyst
Enterprise Architecture
Deloitte Consulting
+31 6 50 07 02 16
mquekel@deloitte.nl

Contents

What is DevOps?

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

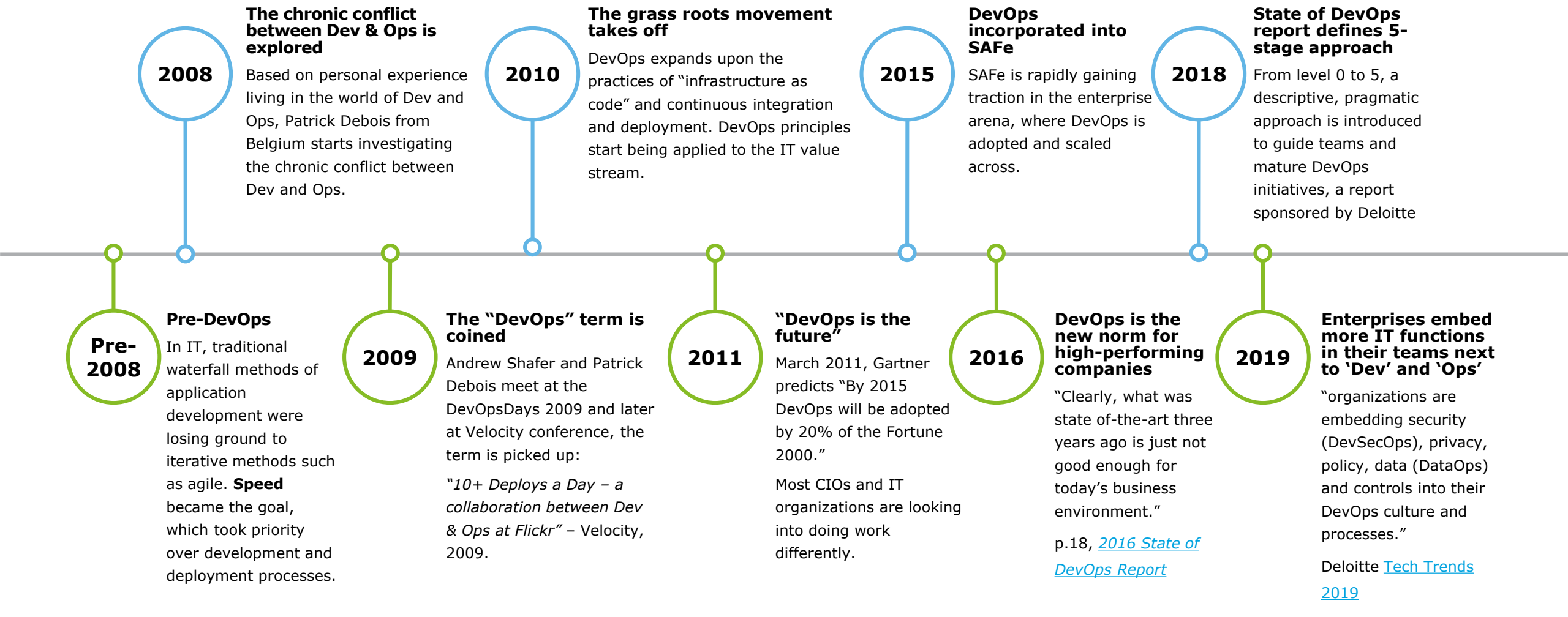
Applying DevOps principles to the SDLC lead to new practices that benefit both Development and Operations

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)

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



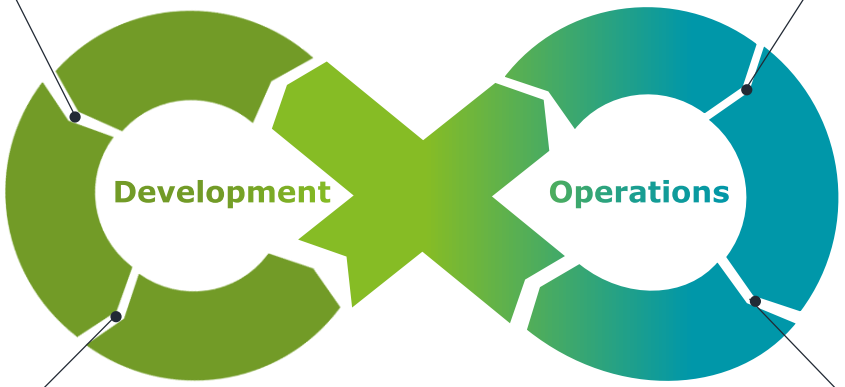
DevOps practices

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

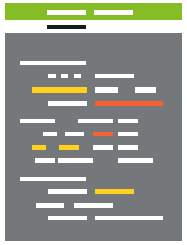
Software Delivery Lifecycle Process



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

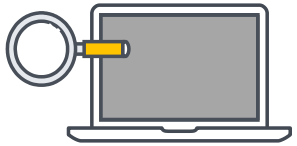
Continuous Testing



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

Continuous Operations

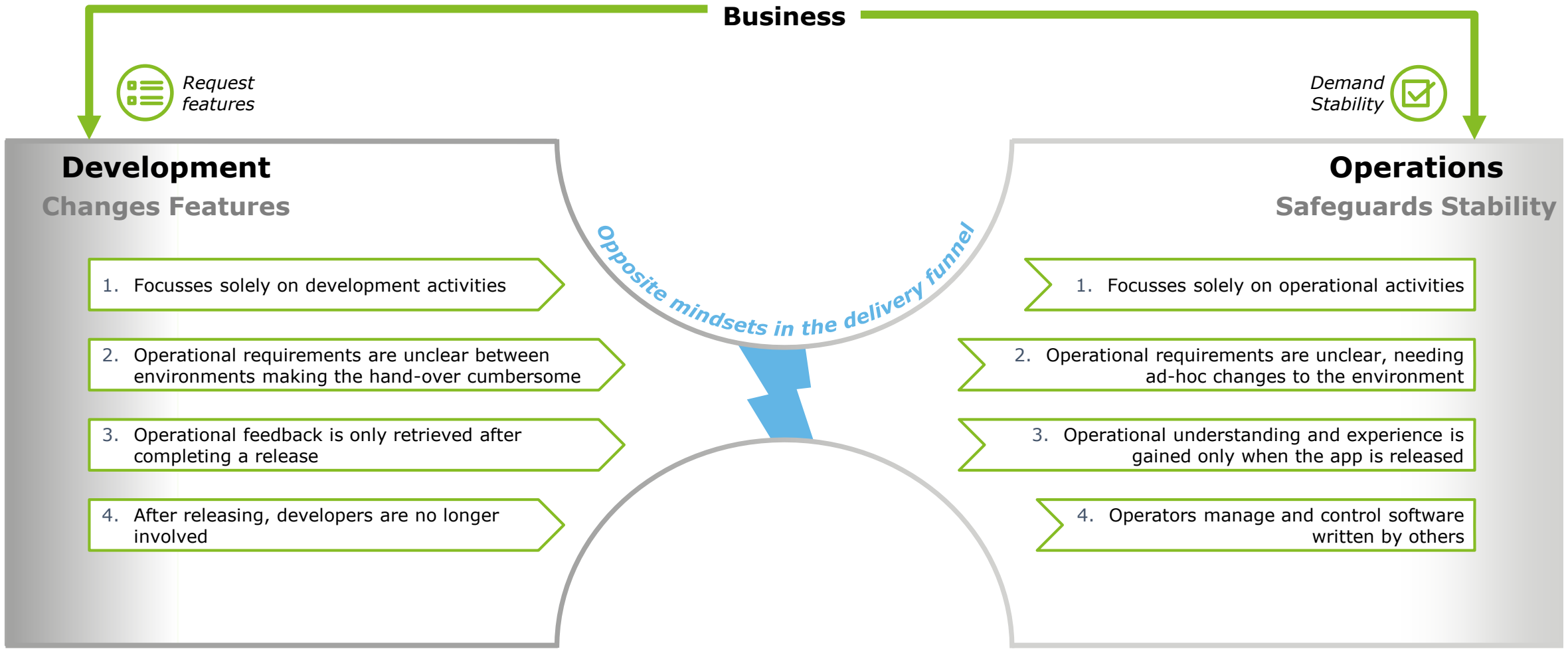
is proactively managing the solution based on feedback loops. Monitoring and telemetry become part of the backlog. Processes such as patching also fall under this practice



Why do I need DevOps?

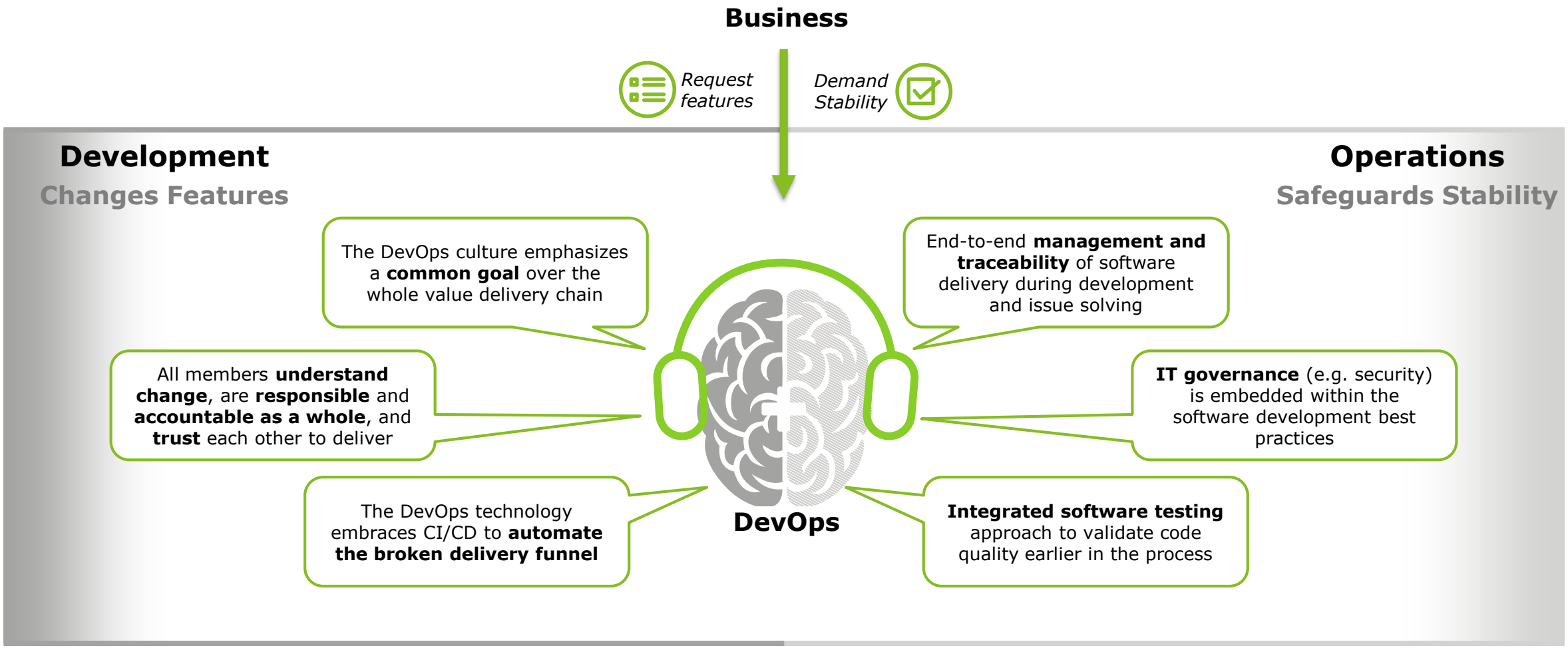
Traditional function separation for Development and Operations

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



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

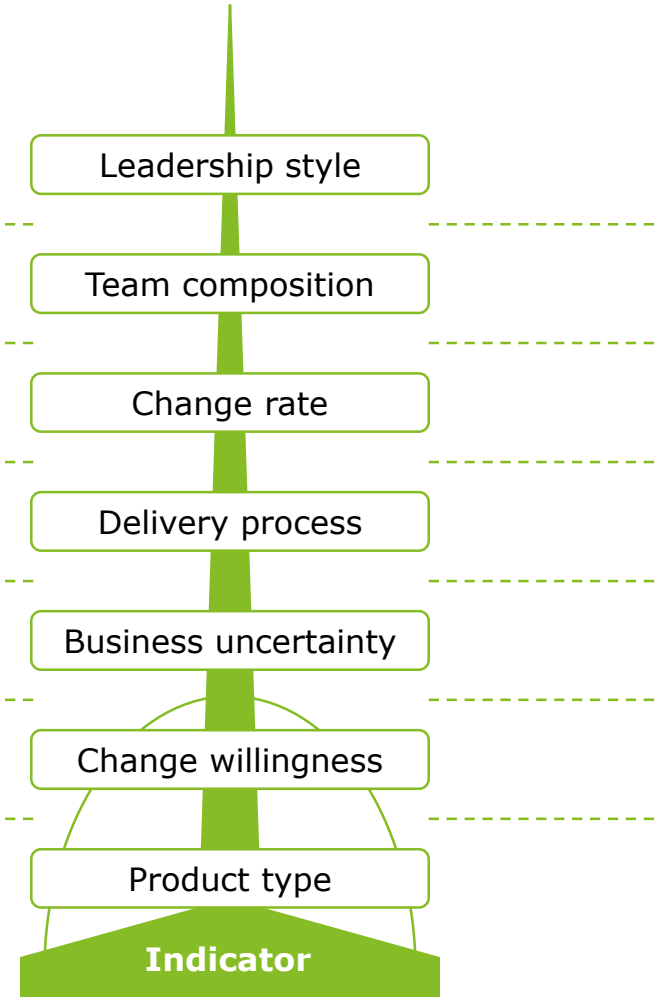


Where is DevOps applicable?

Indicators for DevOps Applicability

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

DevOps is Applicable		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



What is needed for DevOps to work?

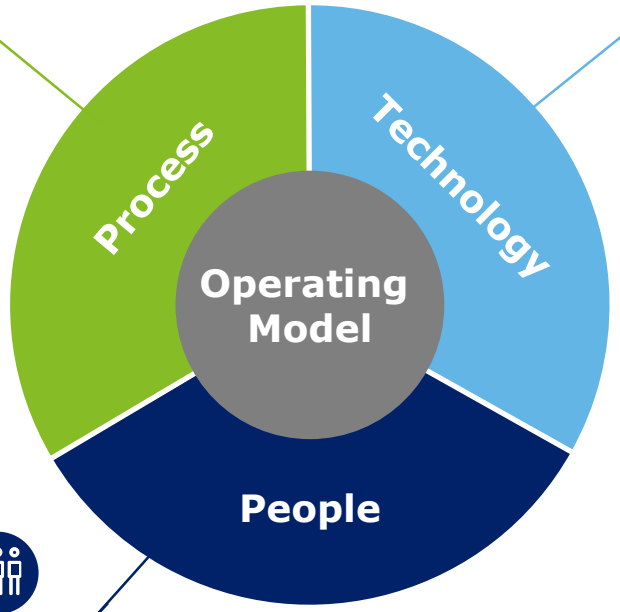
DevOps dimensions

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

People

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 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.

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

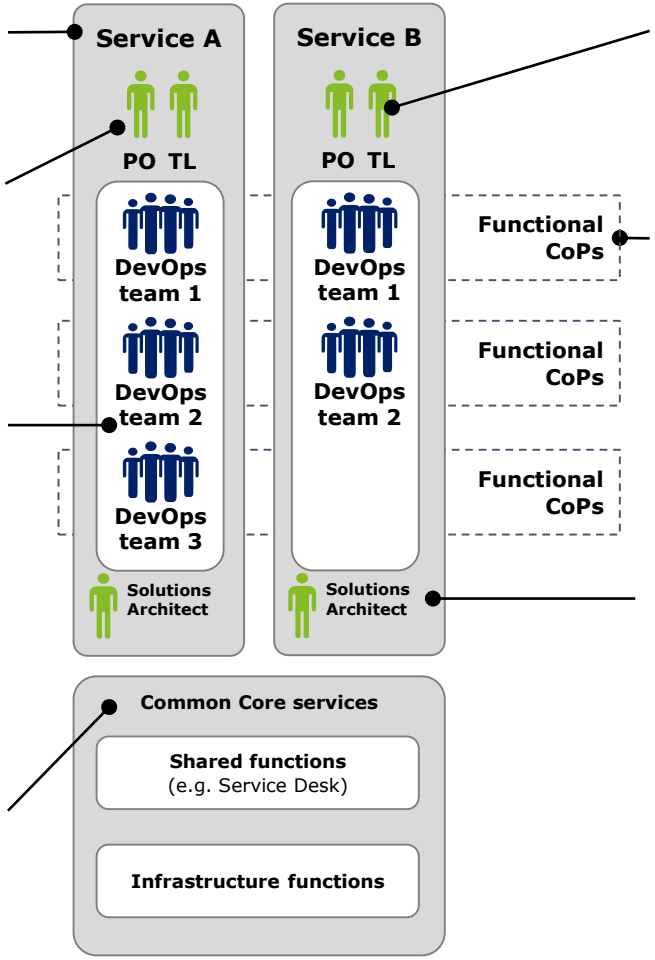
- Culture of shared responsibility and collaboration
- End-to-end ownership of services
- Multi-disciplinary teams
- Incremental value delivery
- Flow optimization in the delivery process
- Automate (almost) everything
- Measurement of everything
- Continuous improvement

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

Product Owner a fixed business resource empowered to shape and direct the development of a product in a way that maximizes business value

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



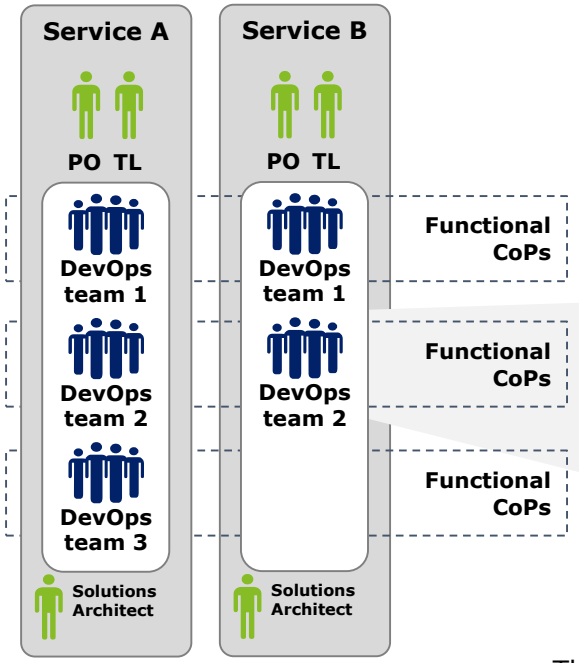
Tech Lead is a product team member with extensive technical development experience who can lead the Product Team in the execution of its work

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

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

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

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

The **Infrastructure Engineer*** configures, maintains and monitors the provisioned infrastructure that is hosting the application (full-stack responsibility)

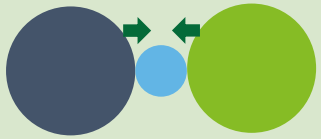
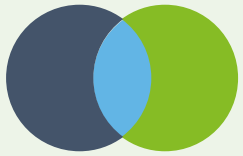
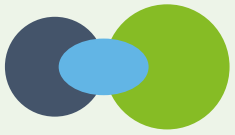
The **Operations Specialist** executes day-to-day technology operations (functional maintenance), monitors technology operations, performs Problem Management, manages change processes (Approves/Rejects)

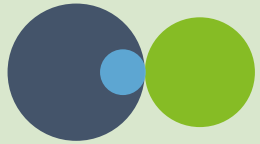
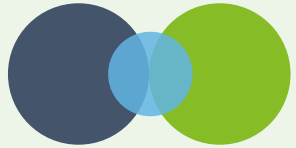
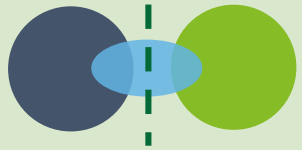
The **Scrum Master** facilitates the team on processes & approach, manages impediments and enables continuous improvement

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

Different team topologies

No one-size-fits-all approach, DevOps can be implemented in many different organizational and team setups

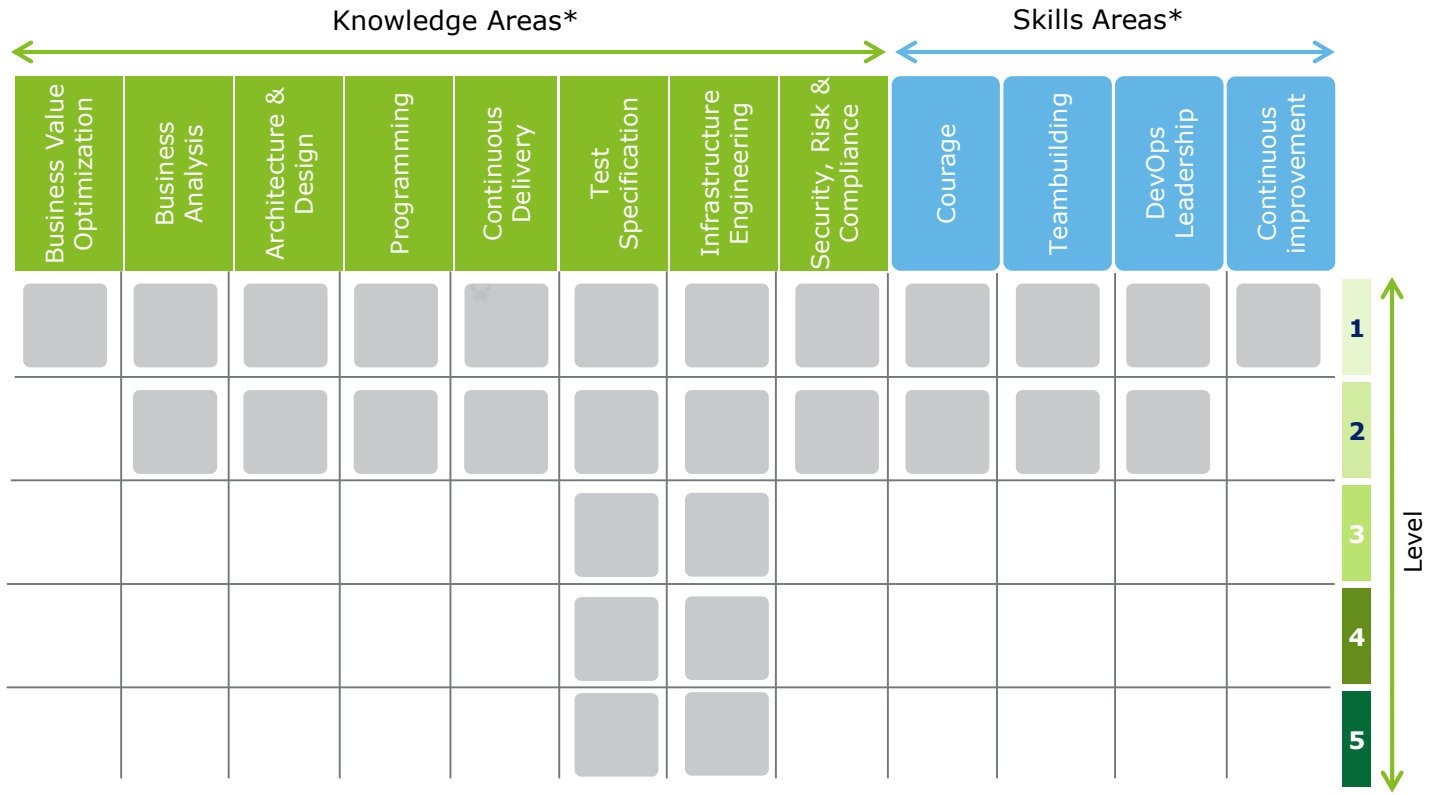
	DevOps setup	Team topology	When to use?
1	**DevOps team with an expiry date** Temporarily setup DevOps team as separate entity to existing Dev and Ops Teams		As an organizational pilot or hybrid state for organizations aiming to adopt DevOps
2	**Dev and Ops collaboration** Enables collaboration and co-creation between Dev and Ops through a common vision and shared responsibilities		To move away from an "us vs them" mindset. (NB: the extent of overlap depends mainly on organization size and resource resources)
3	**Fully shared Ops responsibilities** Fully integrated DevOps team with shared goal and responsibilities		Works best for organizations with a single main product or service
4	**DevOps as an external service** DevOps team supporting smaller Dev teams		Organizations with limited operational issues, DevOps team focusses on supporting dev teams in the problem domains

	DevOps setup	Team topology	When to use?
5	**Ops as infrastructure-as-a-service** Infrastructure operations are fully covered in the self-service model consumed by the DevOps team		Traditional organizations with several products or services in which the Ops teams provides IaaS
6	**DevOps Evangelists Team** Team setup facilitates communication between Dev and Ops teams while keeping the majority of the existing team setup		When the organization is reluctant to change, this setup could be used to slowly transitions towards a DevOps organization
7	**Separated responsibilities for regulated industries** Separate responsibility for Dev and Ops on the DevOps team in order to provide an auditable trail		When organizations report to external supervisory bodies to comply with industry regulations

Development team DevOps team Operations team

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)

The mindset of a DevOps team member

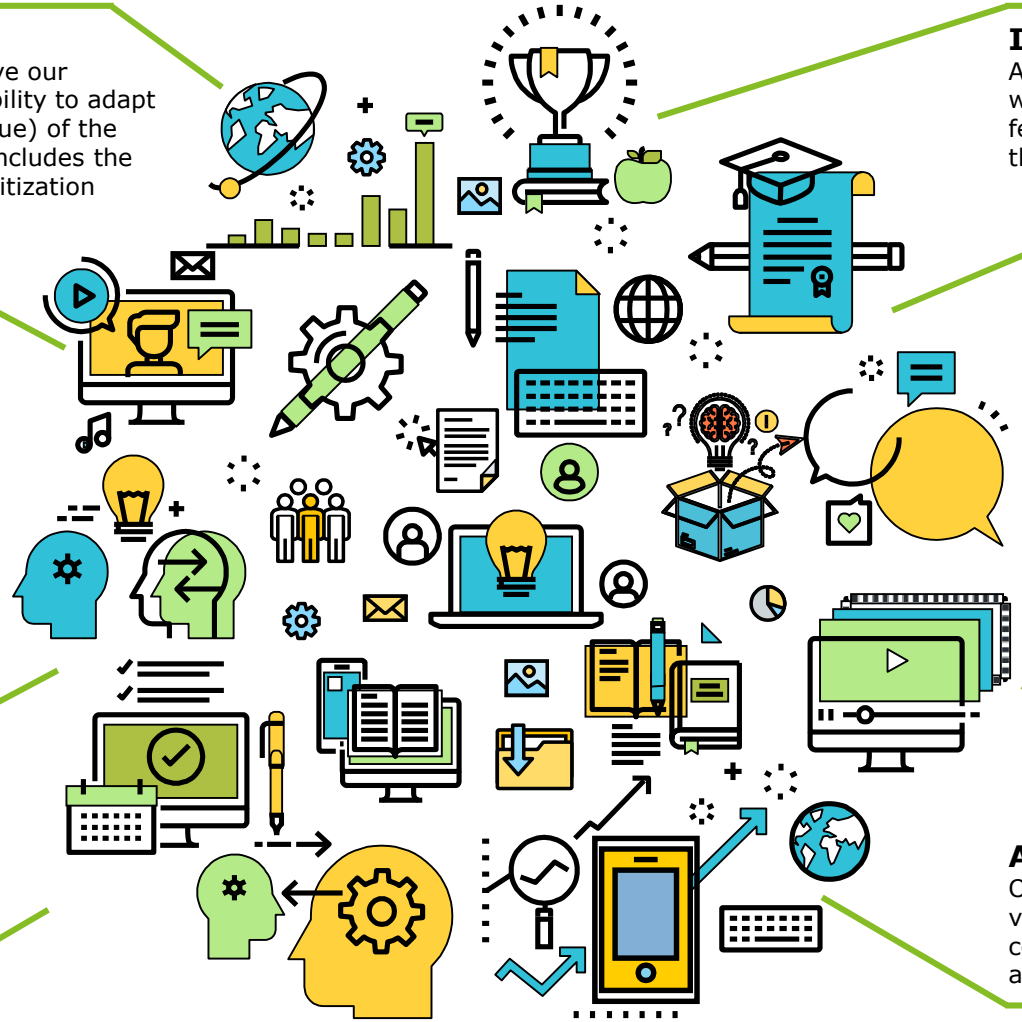
DevOps team members foster certain cultural aspects contributing to the end-to-end ownership of services

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

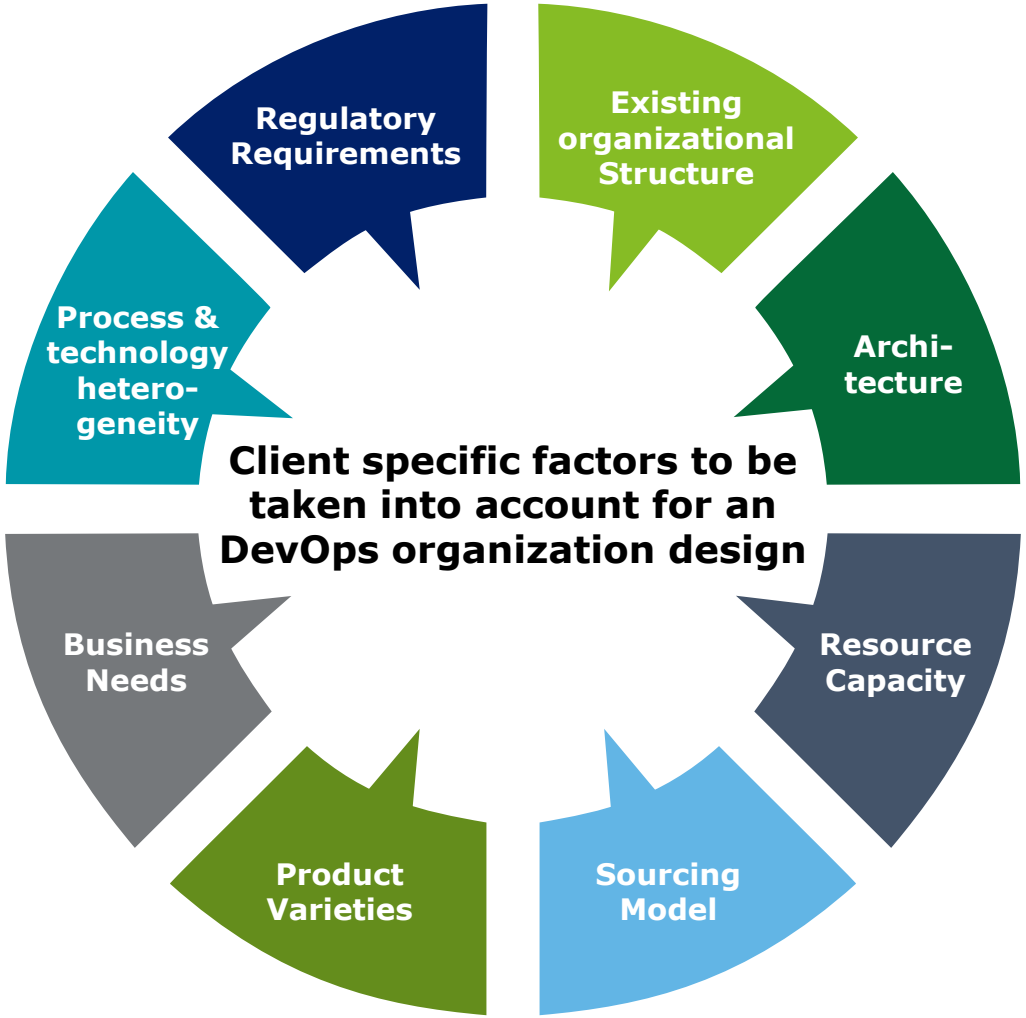
Continuous learning & Continuous improvement
We have the desire to explore and learn in all activities we do. We strongly believe that working together, transparency, and sharing knowledge is vital. We care about our job enough to not pass the buck, we want to learn all the parts as a whole and not just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

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:

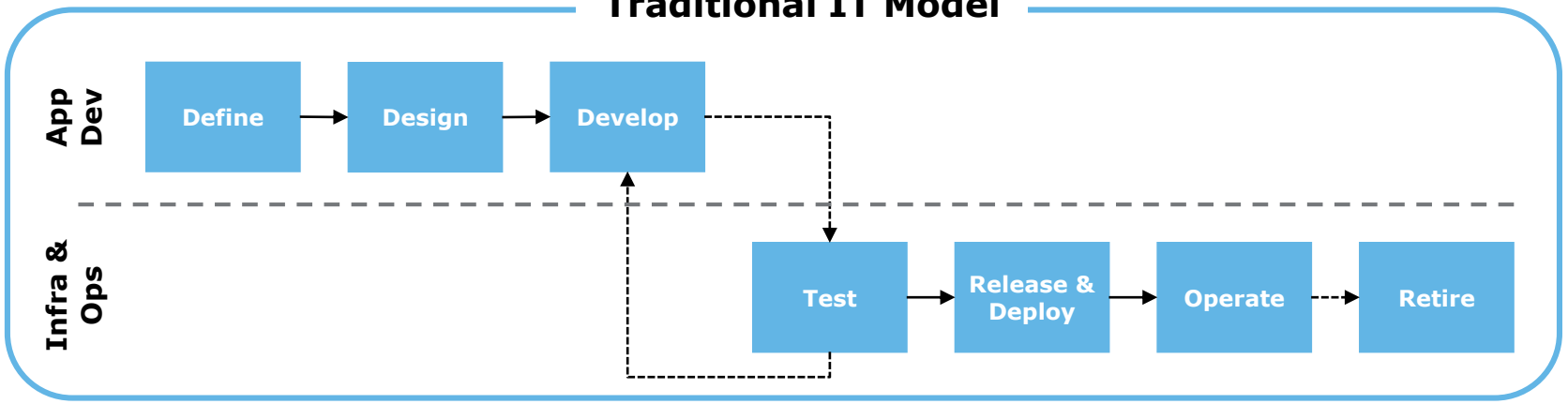
- Keeping some functional hierarchy intact to facilitate collaboration with the enterprise
- Re-architecting the technology stack to enable DevOps practices
- Adhering to some degree of separation of duties to comply with regulations

Process

The DevOps model is significantly different from the traditional IT model

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

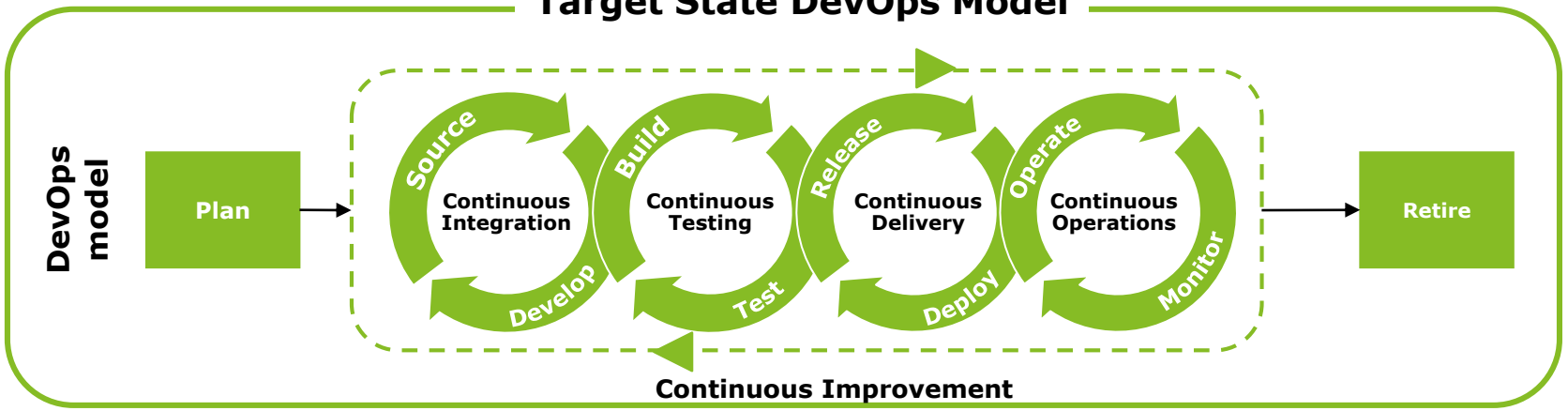
Traditional IT Model



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



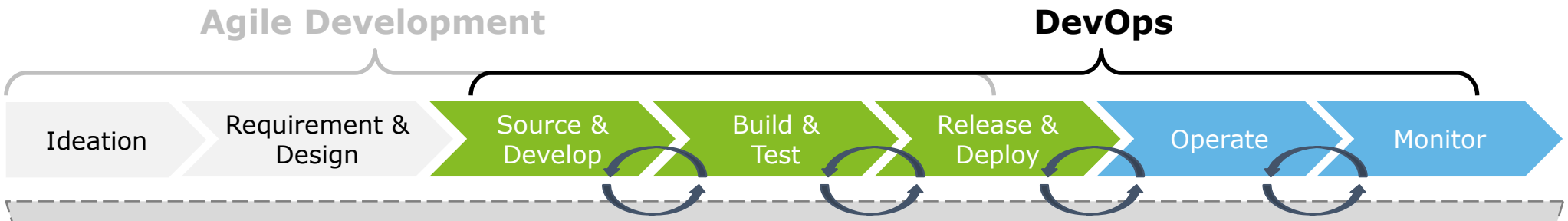
Implications

- 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

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

Technology

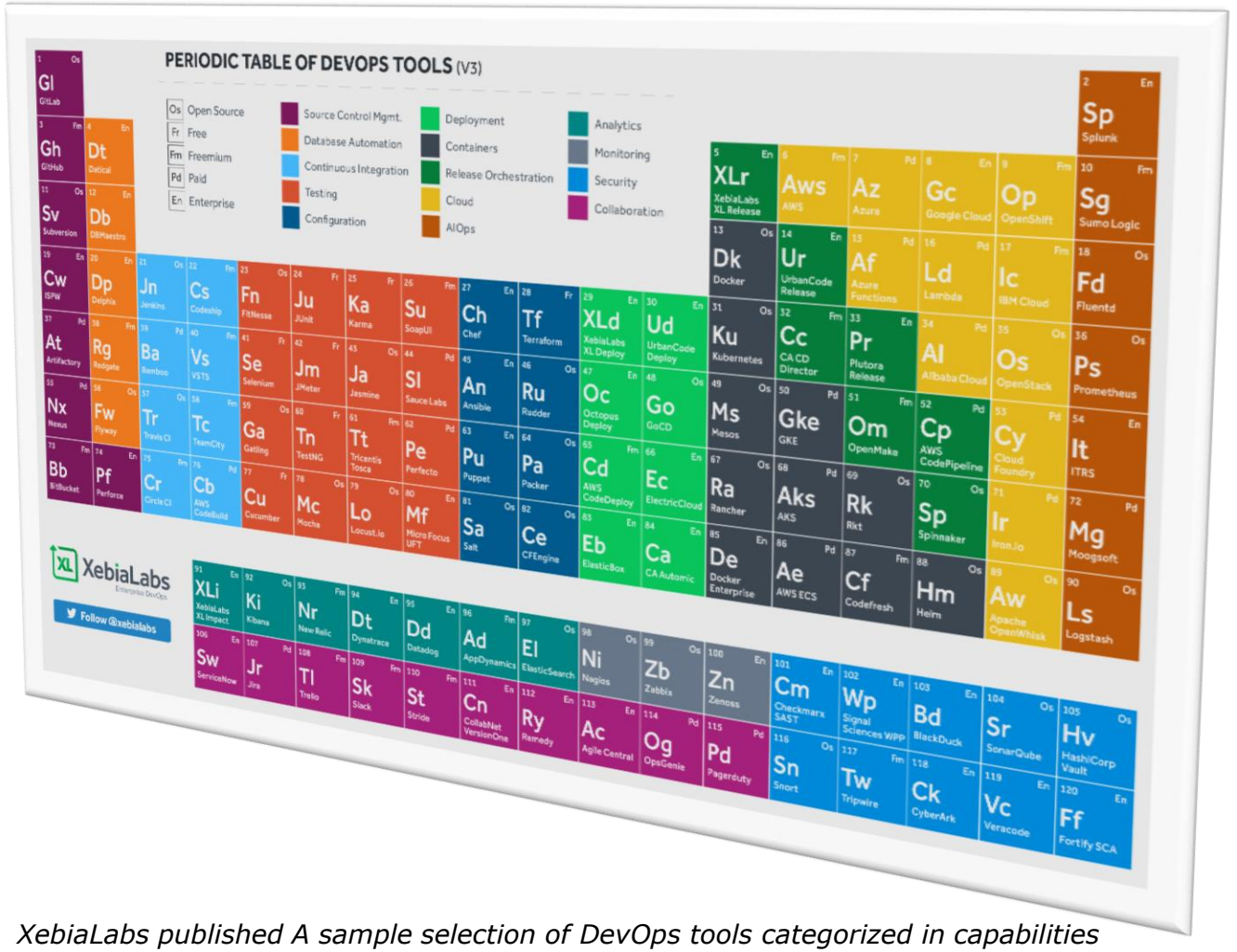
A legion of tools are available to support DevOps practices

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



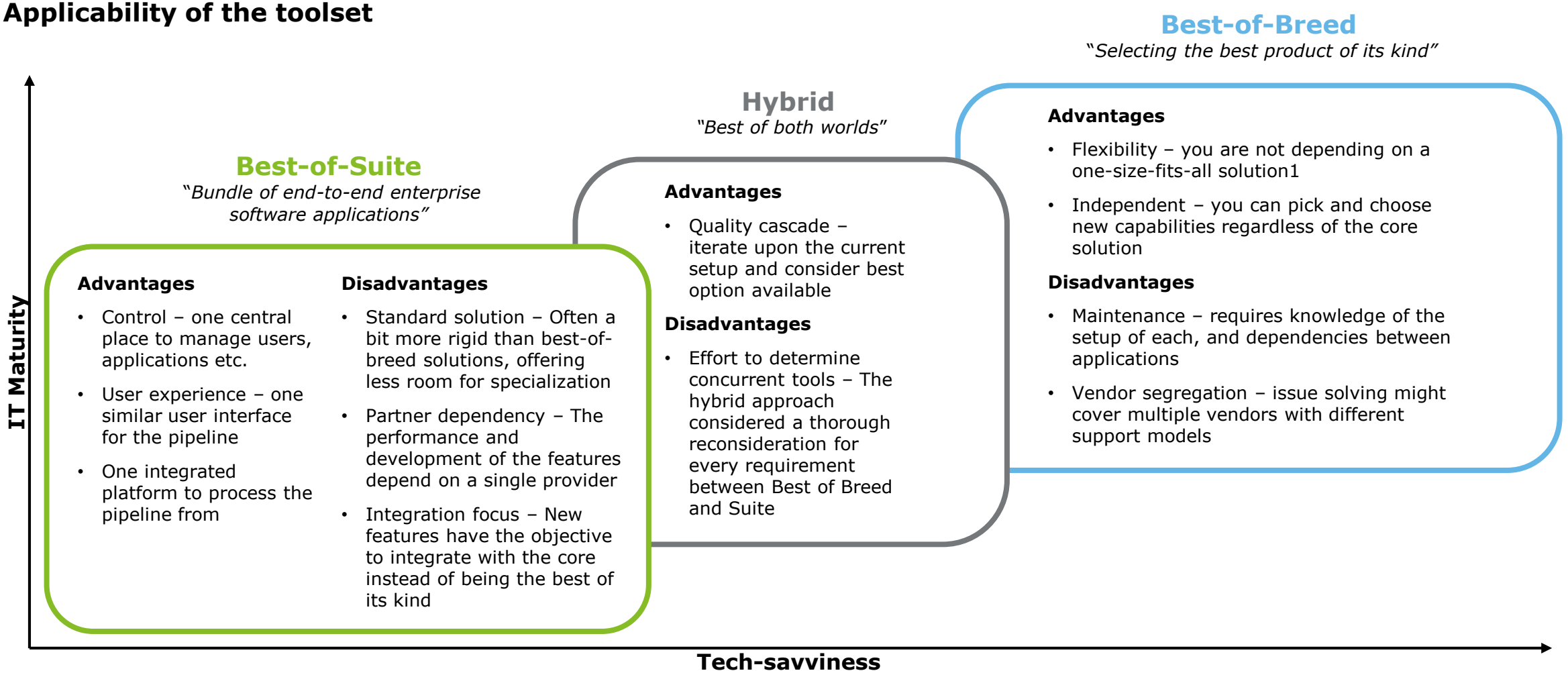
XebiaLabs published A sample selection of DevOps tools categorized in capabilities

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

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



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

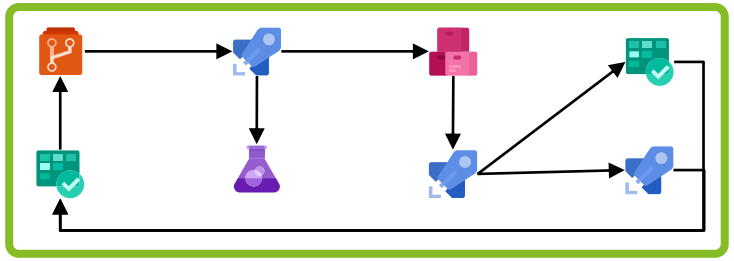
Monitoring the performance of the team in the software delivery process using DevOps metrics and KPIs

Software Delivery Process

Software Operations

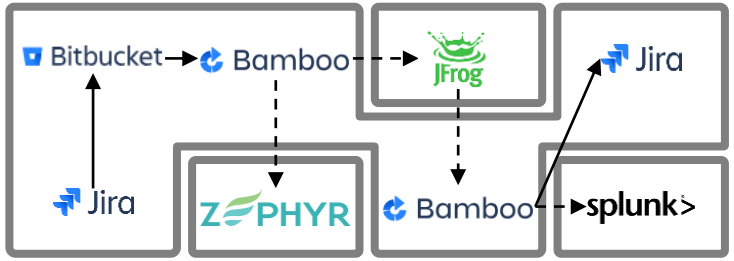
Monitoring of the deployed system's health & performance using application and infrastructure monitors

Best-of-Suite



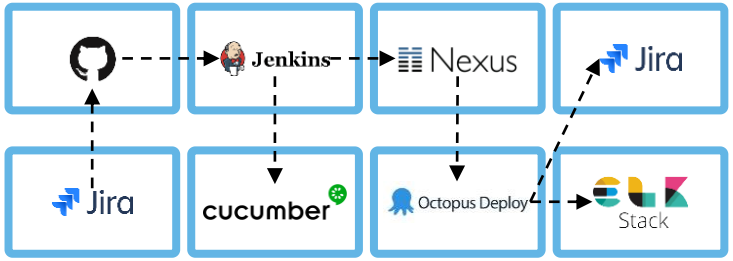
Azure DevOps covers the full extent of the CI/CD pipeline, with no external integration required

Hybrid



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

Best-of-Breed



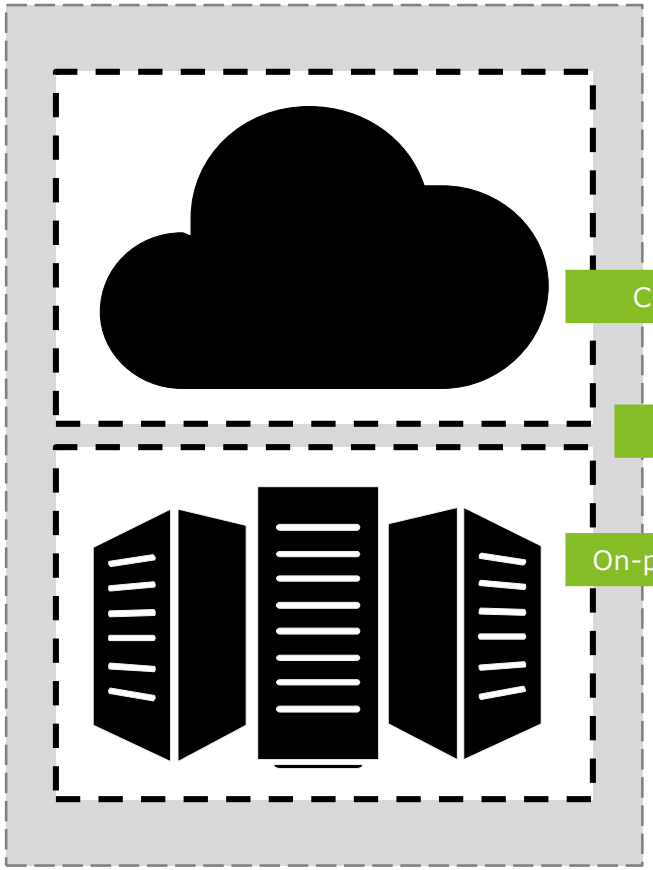
The pipeline orchestrator (Jenkins) becomes the central component to integrate all applications

Same suite, no interface

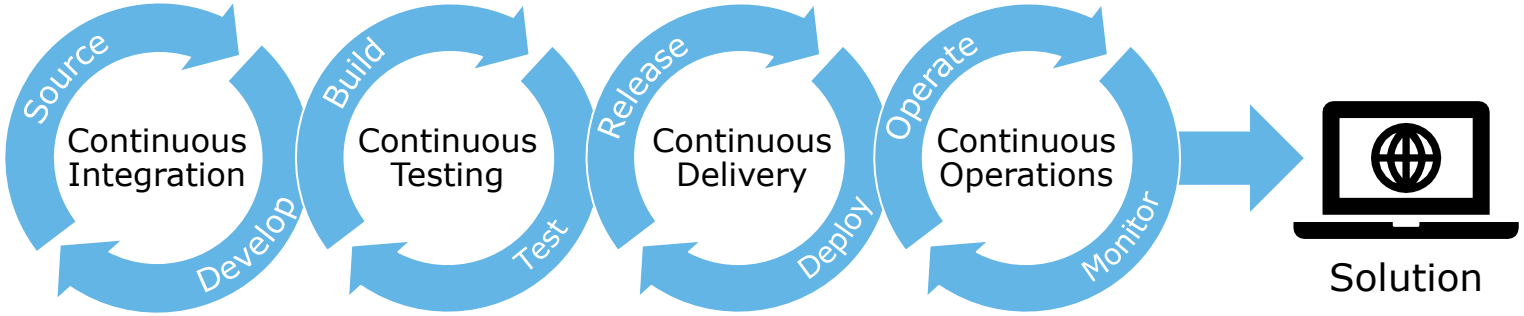
Interface between tools/suites

DevOps and Cloud go hand-in-hand

The DevOps way-of-working makes instrumental use of cloud automation. However, DevOps practices can also be applied in hybrid or on-premise environments



DevOps is accelerated using automation principles that are applied in the Cloud. DevOps methods and tools require a platform that is able to change quickly. Due to the agile and self-service nature of cloud, DevOps teams can collaborate on a single platform, in which they can source & develop, test and deploy new functionalities without the burden of underlying infrastructure.

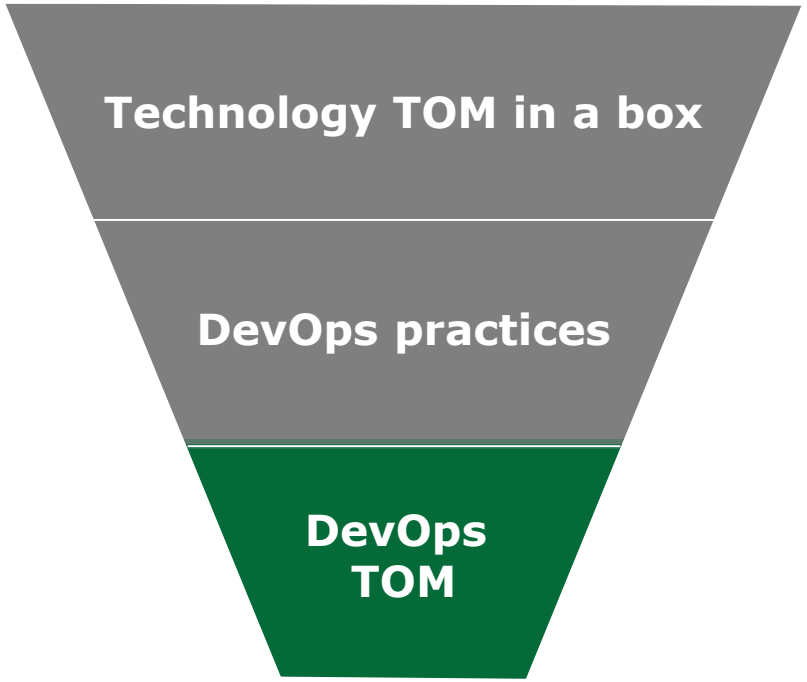


On-premise environments increases the scope of the DevOps teams. Fortunately, DevOps practices are not environment specific and the DevOps teams can add team members with skills required to develop and operate the on-premise infrastructure.

Operating Model

DevOps Target Operating Model (TOM)

The DevOps TOM aims to land 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 an holistic framework to describe the governance structure of an technology 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 organization should govern and operate the DevOps practices

Governance through a DevOps TOM

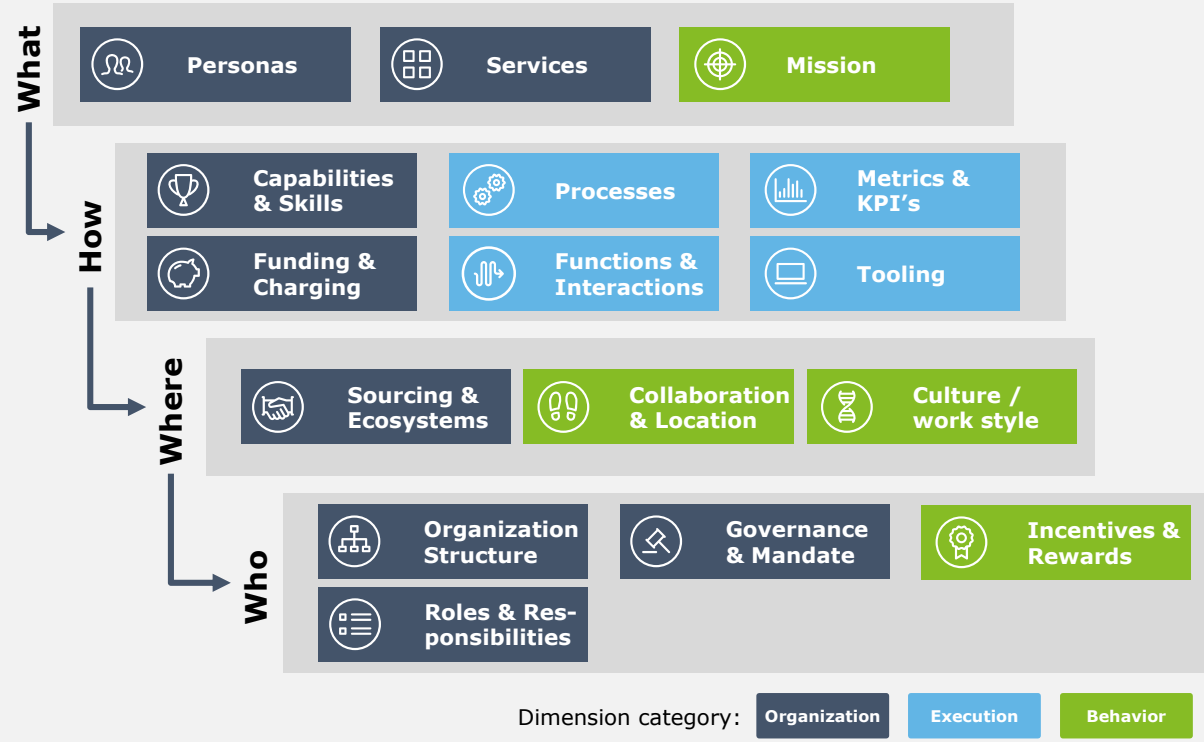
We use the DevOps TOM to explain to a client how it should **Organize**, **Execute** and **Behave** in a DevOps world

DevOps TOM

Using a number of inter-related dimensions we explained to the client:

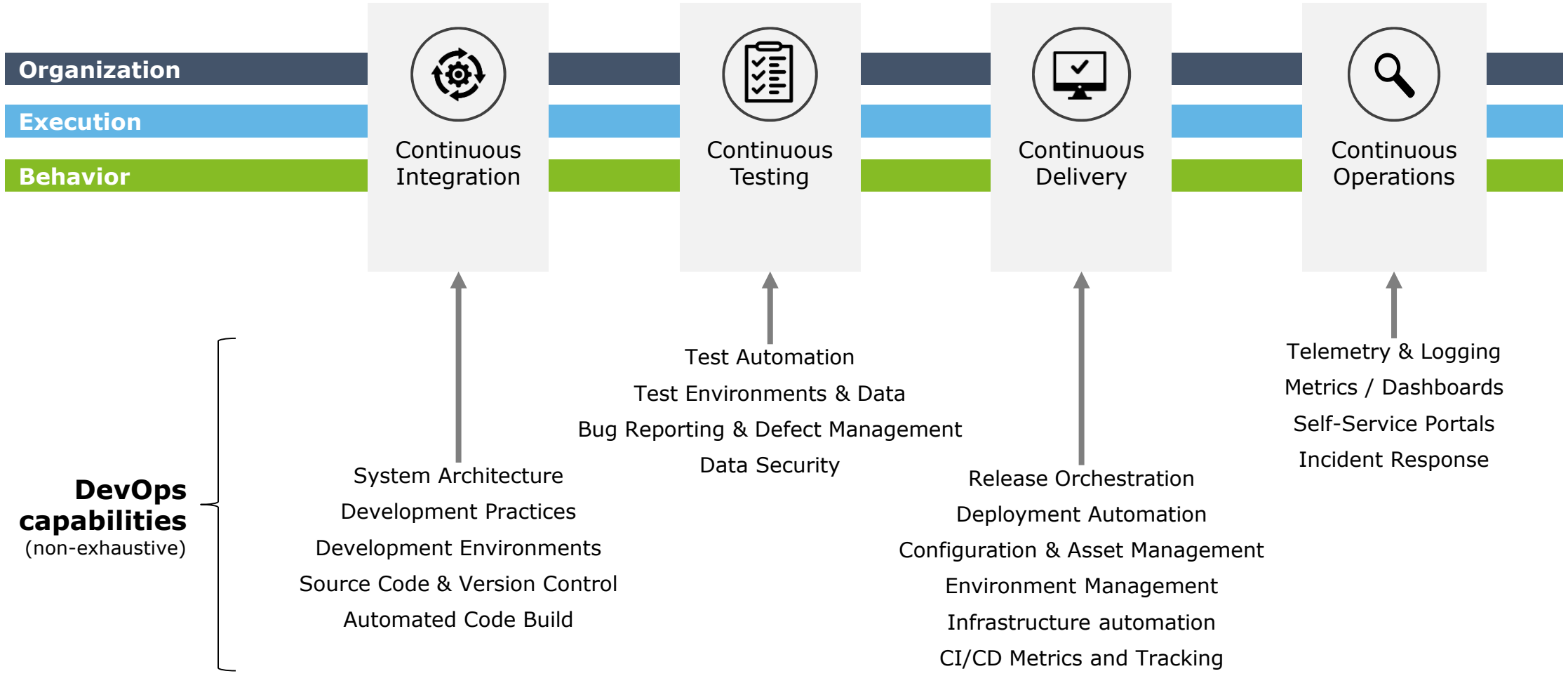
1. ***What*** *services the organization is going to deliver, and to whom*
2. ***How*** *these services are delivered using DevOps capabilities, processes and CI/CD tooling*
3. ***Where*** *these DevOps capabilities are sourced from, what the ecosystem looks like, and how to collaborate in a DevOps culture*
4. ***Who*** *are responsible for service delivery and support using DevOps roles and organization structure*

The Technology TOM describes a number of inter-related dimensions that reflect how a Technology organization is **Organized**, how it **Executes** and how it **Behaves**



Detailing the DevOps practices

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



DevOps operating model differentiators

The DevOps way-of-working has several implications on the operating model that differentiates from traditional operations

Differentiators of the DevOps TOM

Organize

Impact:

- **Service Oriented Organization:** Resources organized around services, focused on delivering designated business outcomes
- **Organize the People:** Creating high performing teams working focusing on a common goal
- **Venture-Capital style Budgeting:** Funding dependent on minimal viable product, and its performance
- **Organize the Work:** Slicing the work into smaller chunks that add value to customers immediately and then investing in it
- **Sourcing Model:** Sourcing model aligned to need of firm based on additional capabilities and delivery methodologies introduced

Execute

Impact:

- **Redefined Roles & Responsibilities**: Redefined managerial roles, integrated within self-organizing scrum teams
- **Iterative and Frequent Releases:** Introduction of DevOps practices for iterative and frequent releases
- **Speed Up the Work:** Providing tools and automating processes to minimize handoffs and maximizes checkpoints to reduce dependencies and risk
- **Improved Interaction:** Siloes broken down between and within the business units and IT organization

Behave

Impact:

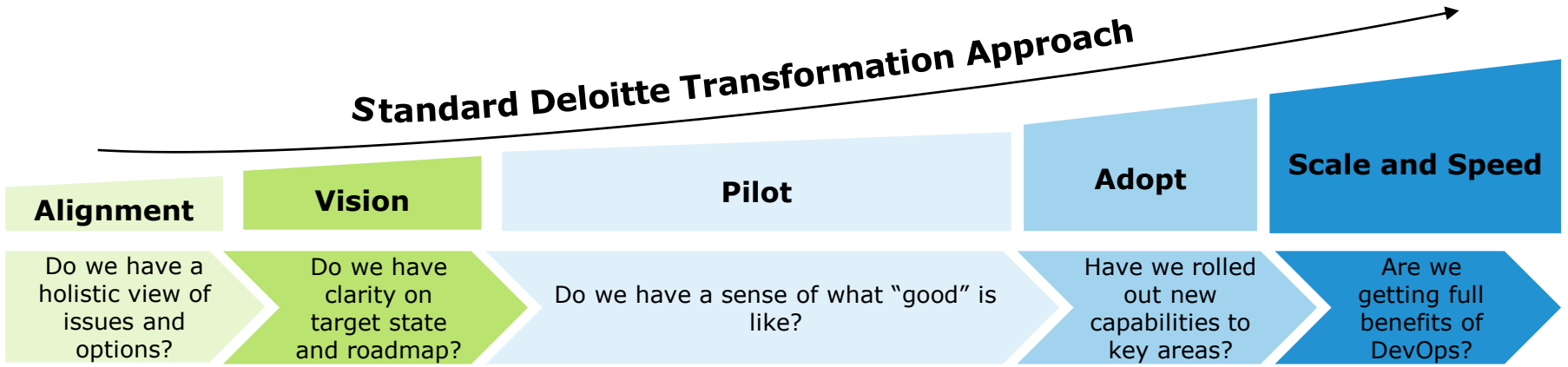
- **Structured Vision:** Vision adapted to changing business needs of the business and customers
- **Collaborative Services & Capabilities**: Increase in usage of collaborative services and capabilities to the support business expectation
- **Visibility and Transparency:** Greater visibility and transparency across the firm with merging of development and support functions and capabilities
- **Cross functional Team:** Resources formed from Run, Change, Design, and Test, to focus on specific product or service which needs to be delivered to customer or business

Source: Deloitte Technology TOM in a box

How do I implement DevOps?

Option 1: Standard Deloitte transformation approach

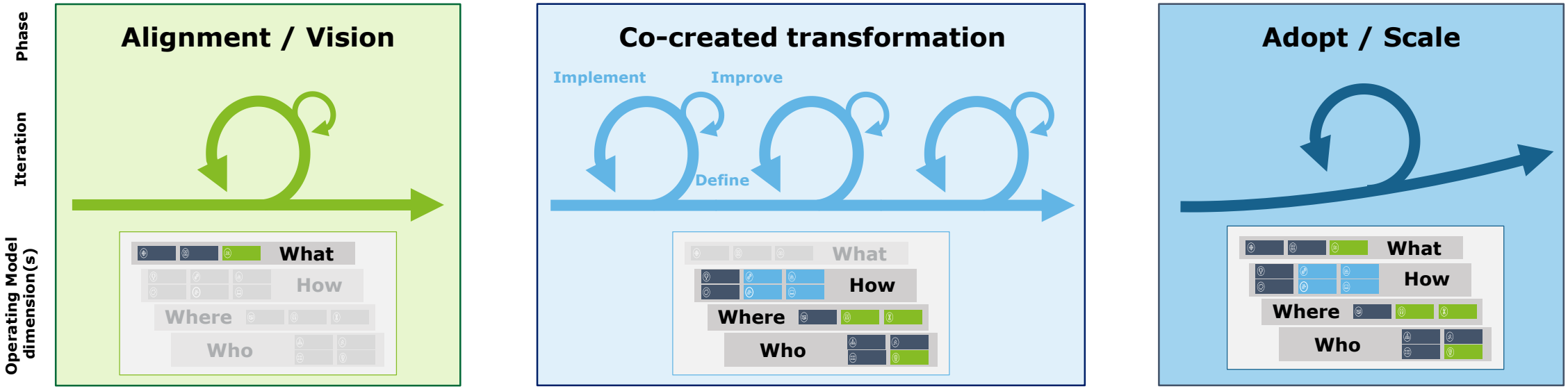
The DevOps transformation journey across a large organization takes 2-3 years, but starts with a clear Alignment & Vision created in 8-10 weeks



Alignment/Vision	Pilot MVP/Targeted Assistance	Adopt/Implementation
Help articulate the vision, execution plan and funding requirements in a short sprint. Deloitte team leads with sponsorship and co-leadership from client team.	Select a pilot to build a MVP with a CI/CD pipeline to establish a model for rapidly identifying and launching projects to propagate DevOps thinking into practice quickly across organization. Deloitte acts as facilitator and starts the transition as the client takes on a more coaching/managing role.	Establish Nerve Center/Center of Excellence for scaling DevOps capability. Typically done in a Build Operate Transfer (BOT) model.
8-10 weeks	8-12 weeks	18-24 months

Option 2: Co-created transformation

A future vision stays the basis for the DevOps TOM, but for delivery we can shift towards a bottom-up approach and co-create change iteratively with the client



Imagine the future of services in the service organization (dot on the horizon):

- What is the mission of service organization?
- What IT services, and to whom, is the service organization going to deliver?

Co-create the DevOps Operating Model iteratively using a combined Deloitte client project team:

- Each iteration defines and implements a prioritized set of DevOps capabilities based on the delivery of CI/CD pipeline technology
- Scope is determined prior to each sprint based on:
  1. Actual CI/CD pipeline technology delivered
  2. Relevant Technology TOM in a box dimensions
- Collaboration with the client:
  1. Full support and dedication of DevOps champions (dedicated client project team members)
  2. 2-wk alignment with internal client stakeholders

Scale the DevOps Operating Model and technology within the client organization depending on the transformation need.

Establish Nerve Center/Center of Excellence for scaling DevOps capability.

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

1. **Skill Challenge:** Keeping pace with the required skills may challenge your team
2. **Scarce talent:** Some special skills in your organization can’t be replicated for every DevOps team
3. **No magic:** DevOps will not compensate for potential lack of responsibility in your organization
4. **Lack of Overview:** Progress and stability are spread across the teams and overview may lack
5. **Dev Slowdown:** Operations may hamper progress in development
6. **Self-organization Challenge:** Clear Service-Level Structures in operations may be challenging
7. **Management Challenge:** DevOps can mean management challenges for your team leads

Key Lessons Learned

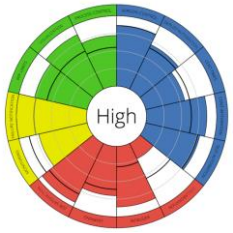
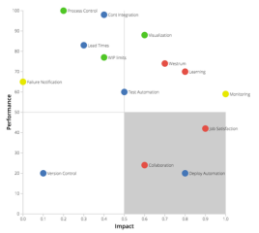
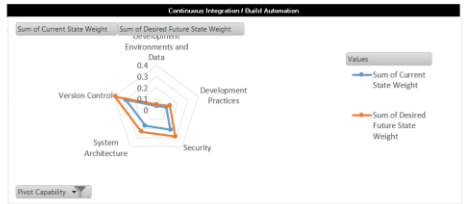
During our engagements we gathered the following key takeaways that we will bring to future projects

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

Deloitte Accelerators

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**

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

Client example 1

DevOps journey and CI/CD pipeline implementation

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

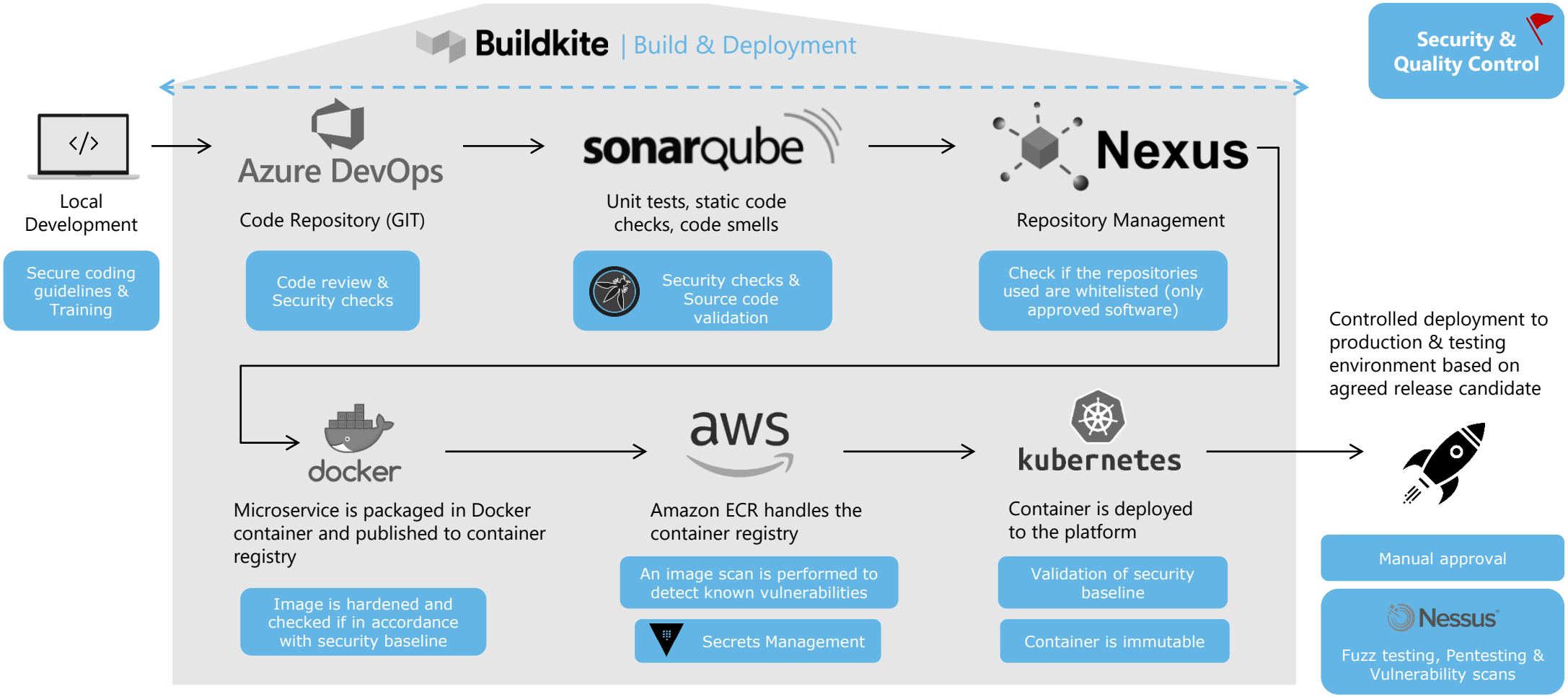
- Gather data from CI/CD process
- Infrastructure logging & monitoring of develop, test and production environments
- Operational data logging, monitoring and analytics on operational process execution
- Provide dashboards to view and report on performance

Client example 2

A CI/CD Pipeline for a Banking Platform

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

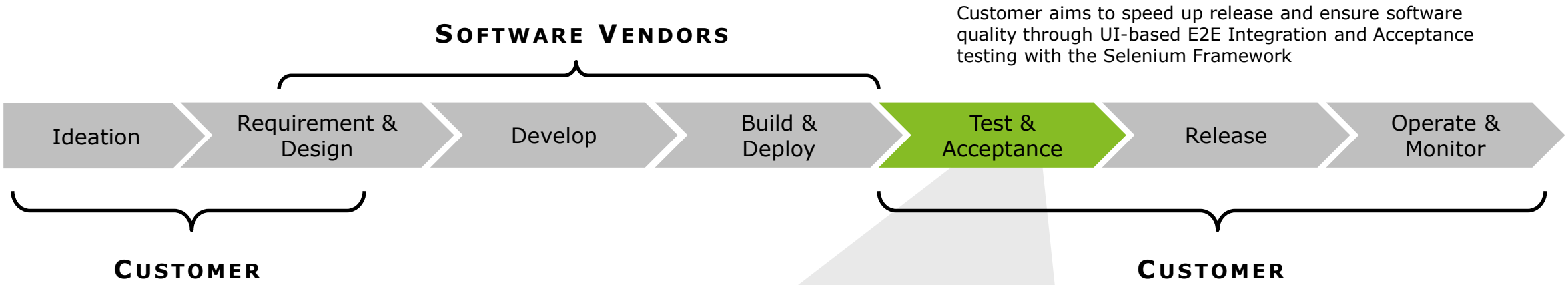


Client Example 3

Test Automation for a Banking Application

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

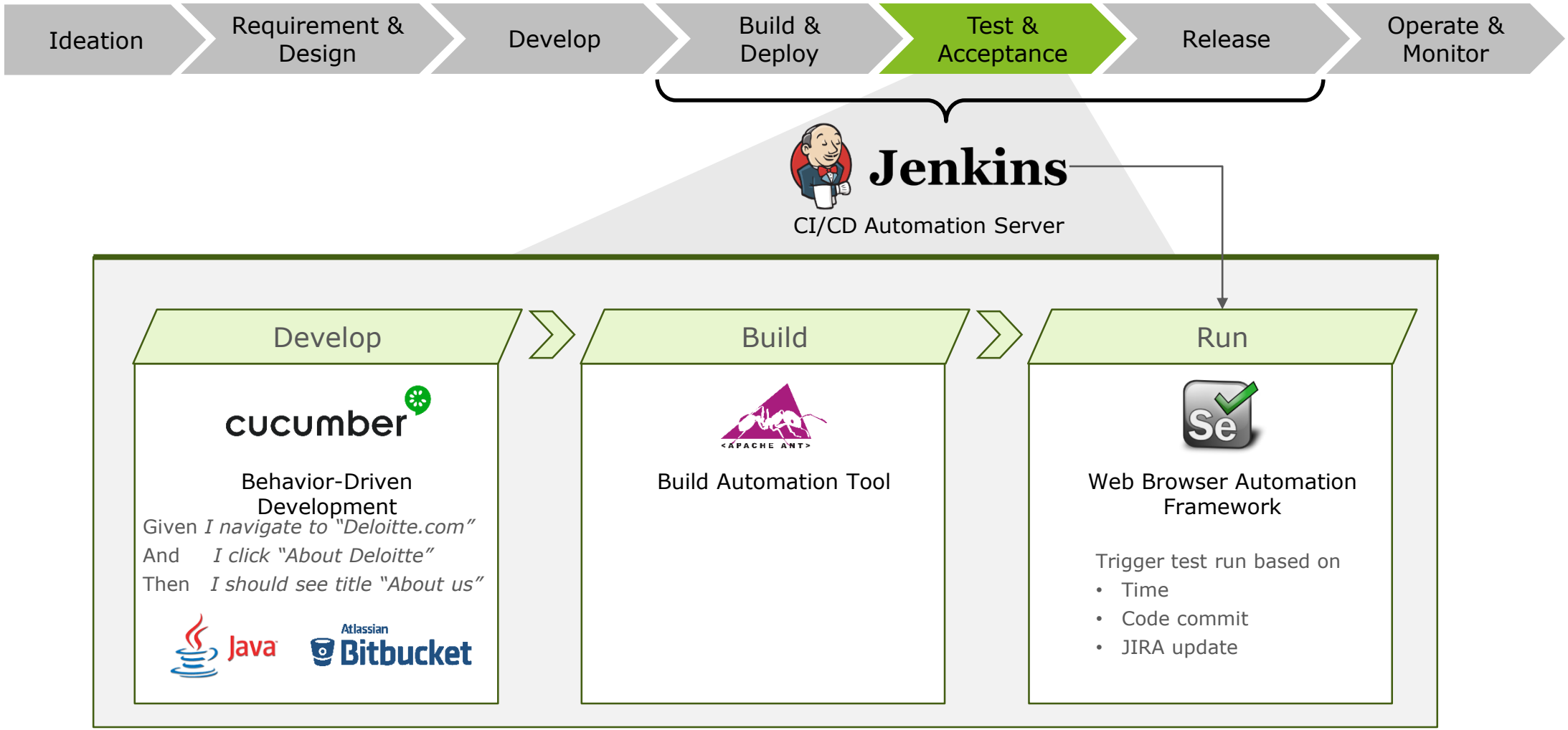


Test Phase	System	Integration	E2E	Acceptance	PAT

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

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



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

Appendices

Additional contributors

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

Andries van Dijk
Director
IT Strategy
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
Enterprise Architecture
Deloitte Consulting
+31 6 83 33 96 09
aboon@deloitte.nl

Tom Pennings
Consultant
Enterprise Architecture
Deloitte Consulting
+31 6 22 17 87 17
tpennings@deloitte.nl

Jan Prüst
Consultant
Enterprise Architecture
Deloitte Consulting
+31 6 82 51 86 04
jprust@deloitte.nl

Audrey Sie
Analyst
Enterprise Architecture
Deloitte Consulting
+31 6 50 06 26 13
ausie@deloitte.nl

Jasper Lelijveld
Analyst
Enterprise Architecture
Deloitte Consulting
+31 6 50 06 25 77
jalelijveld@deloitte.nl

Deloitte.