EBOOK

A Practical Guide to DevOps for SAP®

basis

DevOps + Testing for SAP

DevOps and its impact on SAP

In today's fast-paced world, business leaders are under extreme pressure to innovate in order to deliver value and increase competitiveness. The role of enterprise applications in supporting these goals can't be underestimated — such systems need to have the agility to adapt and respond safely to frequent, rapid change.

If you want to deliver more value to your customers and be part of a high performing organization that thrives in a fast-moving digital economy, you should consider how embracing DevOps for SAP could help.

Without DevOps, deploying even small changes to SAP systems can take a huge amount of time, due to out-dated development and test processes, long release cycles and concerns about risk and stability.

However, SAP customers who adopt DevOps and continuous delivery can take advantage of shorter development cycles, which translates into faster realization of fixes and improvements to their business, resulting in faster time to market and the ability to differentiate themselves from their competitors.

Before adopting DevOps it's important to understand that it is more than a technical issue — an evolution in culture, process and tooling is required. This was discussed in another Basis Technologies e-book, "DevOps for SAP: Laying the foundations for success."

In this white paper we will explore the practicalities of implementing DevOps for SAP systems and the impact this approach can have at the various stages of change delivery — from development, through testing, pre-production and production.

The problem of the now

Traditional SAP delivery processes that don't include DevOps or agile development can suffer from a number of key business issues, including:

- Slow application and change delivery, which results in business frustration and missed market opportunities
- Deployed applications not meeting business needs, due to lengthy development and testing processes and the inability to adapt to rapidly changing requirements
- Risk of downtime and outages when revisions are released into production.

These issues are typically caused by a number of common contributing factors, including:

- Disconnected teams that operate in silos and do not integrate business users into the process. Typically work is passed between teams with little ownership, communication or visibility, which leads to low levels of satisfaction
- Highly manual and error prone processes for development, approval, deployment and testing
- An inability to understand dependencies between requirements that results in long and slow release cycles
- Use of spreadsheets to manage SAP transports, which causes incomplete or out of sequence deployments and errors in downstream QA and production systems
- A lack of understanding of development requirements and poor or non-existent quality control processes that result in high levels of rework and add unnecessary costs and delays
- An inability to provision relevant "production like" QA systems for testing
- Predominantly manual and lengthy regression test processes that are costly and delay releases, resulting in inadequate (or non-existent) regression testing and a reduction in confidence in the safety of new feature releases
- Lack of visibility into who is doing what and in which systems. In a distributed multi-partner environment, which is a common scenario, this can lead to poor efficiency and large overheads

Agile:

An agile approach combined with the implementation of DevOps can solve these issues from both a cultural and process perspective. However, it is vitally important to understand what the impact will be on the end-to-end SAP development lifecycle.

DevOps as a solution

In order for DevOps to address the issues associated with traditional manual SAP delivery processes, there are some key concepts that need to be understood.

The term “Shift Left” has been coined in recent times and is fundamental to DevOps. It advocates that teams should focus on building quality into their applications at the very start of the process in development.

Equally critical is the ability to release the enhancements that are vital for the business as soon as they are ready. This is often problematic in SAP environments where locked-in release cycles prevent delivery outside agreed timescales.

Automation is therefore a key enabler of DevOps, driving the process improvements that are needed to support modernized application delivery.

DevOps and continuous delivery combine speed, innovation and stability with high levels of automation to deliver efficiency, predictability and consistency, even when deploying high volumes of change. Continuous integration, deployment and testing allows issues to be identified early in order to reduce time and cost and improve quality. And smaller, more frequent releases reduce risk and improve recovery times.

It’s important to understand that continuous delivery doesn’t mean that every change is deployed to production as soon as possible. It means that every change is proven to be deployable at any time as needed.

DevOps consists of a number of ‘continuous’ processes. Key areas in DevOps for SAP include:

Continuous integration

Automation of unit tests, code checks, build sequence, dependency checks and environment provisioning to improve quality and reduce risk.

Continuous delivery

Automation of approvals, testing and delivery for more frequent and safer software releases.

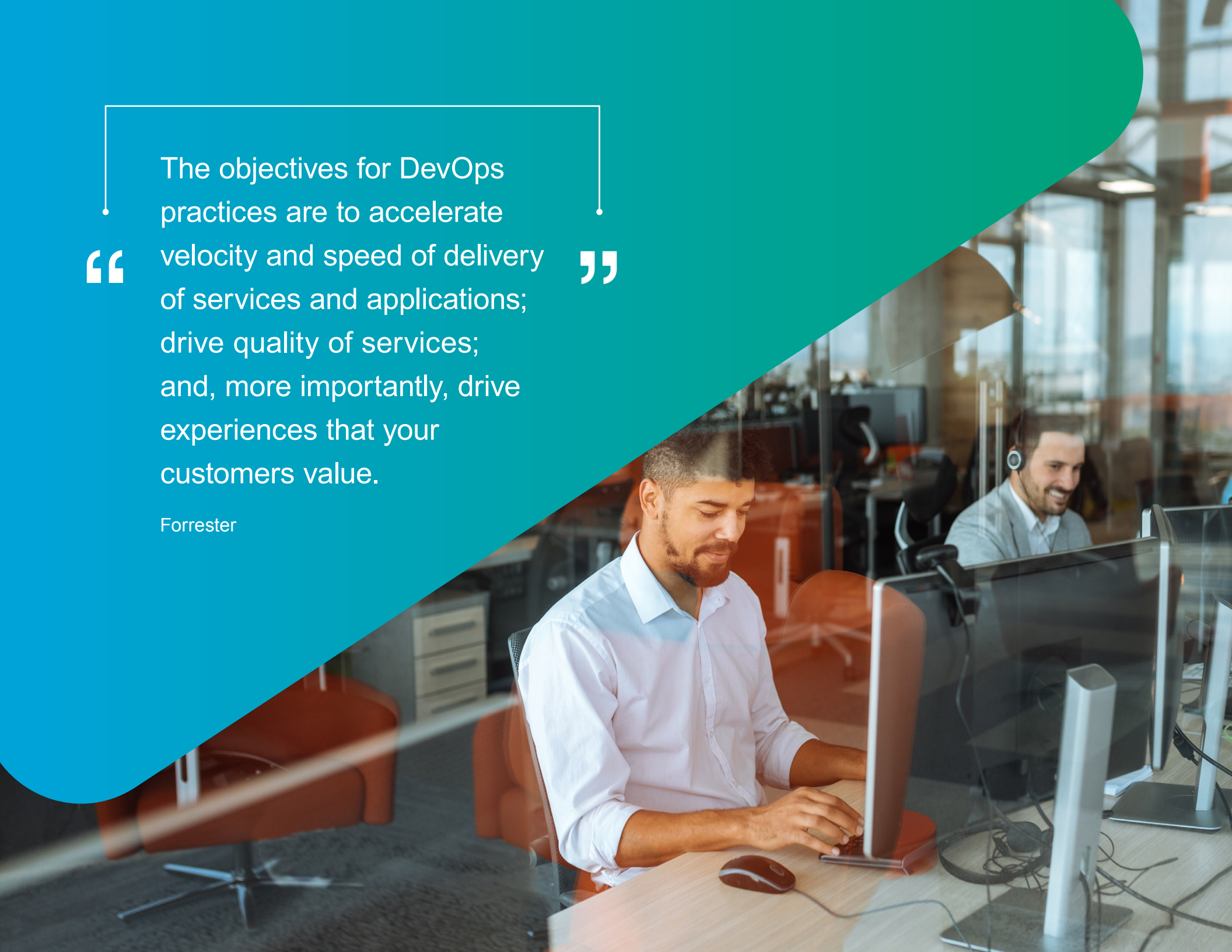
Continuous testing

Automation of testing to accelerate releases, improve confidence and reduce risk.

Continuous deployment

Automation of the release and deployment of changes across both SAP and non-SAP systems.

> The objectives for DevOps practices are to accelerate velocity and speed of delivery of services and applications; drive quality of services; and, more importantly, drive experiences that your customers value.
>
> Forrester



Impact of DevOps on the SAP development lifecycle

DevOps means different things to different people and how it is applied will alter depending on the level of maturity of a given organization and its current processes.

The most common method for delivering new SAP features and functionality follows a process from change request to delivery and review.

Typically, ITSM tools like ServiceNow or BMC Remedy are used to manage the lifecycle of a change but in addition, further tools and processes are required in SAP to enforce governance and audit rules, and to manage workflows and approvals.

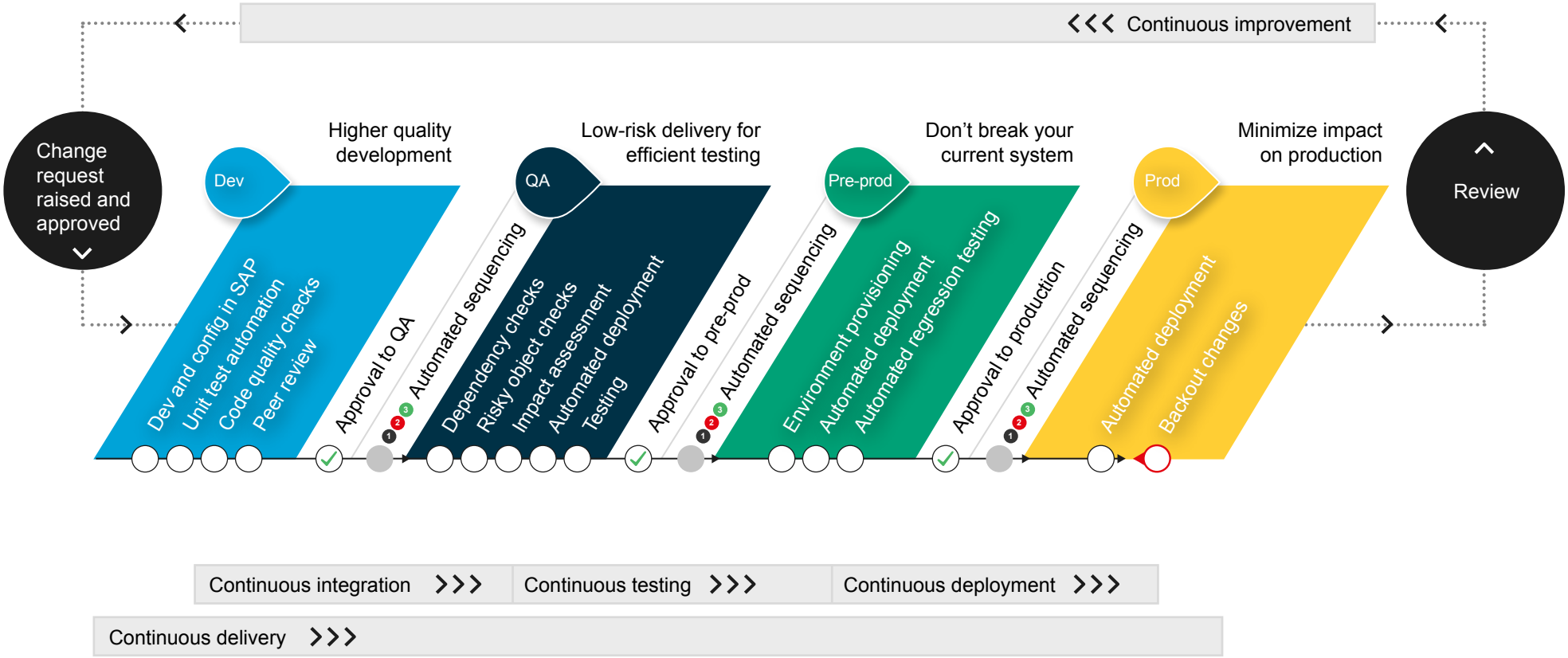
Central to the overall process is a traditional cycle that includes development, QA, pre-production and production. In the following chapters we will look at some practical steps organizations should take when adopting DevOps in order to introduce continuous integration, testing, deployment and delivery into these systems.

Many of the steps proposed rely on automation — a critical part of any DevOps approach. Whilst DevOps concepts are universal, common automation tools are not suitable for use in SAP environments because of its specific architecture. As such, a realistic approach to the areas discussed will require specialized tools such as Basis Technologies' DevOps and test automation platform.

> In response to the rapid change in business today, DevOps can help organizations that are pushing to implement a strategy to support their digitalization efforts. Digital business is software (and technology) — this means that organizations that expect to thrive in a digital business environment must have an improved competence in software delivery.

Market Trends: DevOps — Not a Market, but a Tool-Centric Philosophy That Supports a Continuous Delivery Value Chain.
Laurie F. Wurster, David Paul Williams, Thomas E. Murphy; Gartner, October 2016

The SAP DevOps development lifecycle



Higher quality development

Organizations should not underestimate the importance of building quality into the development process. Doing so enables potential issues to be highlighted at the beginning of the process (Shifted Left) rather than later on in QA or worse, production, where they are much more costly to rectify.

What you need for DevOps for SAP

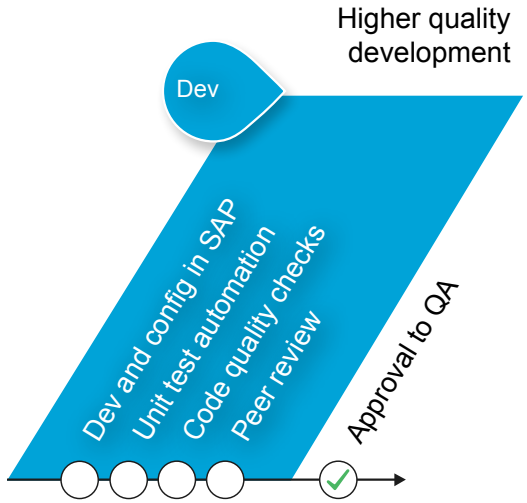
Delivering at pace requires a robust development process that combines business requirements (via product owners) and delivery with constant feedback so all stakeholders and interested parties understand and have complete visibility of what's going on.

How you can do it

- Ensure that objects that shouldn't be changed are either locked down or highlighted for further action or checks. This is particularly important in a templated environment where global and regional changes need to be controlled and developers and configurers are only allowed to make changes in the designated systems. Managing this at the start of development can help to remove a lot of issues and knock-on effects later in the lifecycle.
- Introduce daily stand-up sessions together with planning and retrospective meetings, to ensure that all team members are aligned. Everyone involved in the SAP change process should attend, including functional product owners, development, QA, operations, even the security team. This approach addresses the main risk with distributed teams: communication breakdown.

  Stand-up sessions:

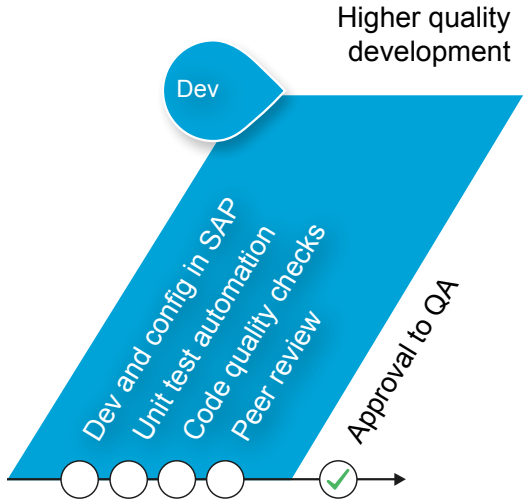
  - Allow people to understand priorities and dependencies, provide constant communication and checkpoints and enable risks to be managed more effectively
  - Maintain visibility of change and development activity across the entire SAP estate so you know who's changing what, in which systems and what the current status of activity is
  - Break barriers down so there are open channels of communication and collaboration between teams. Having development and QA working more closely creates more robust and relevant solutions.



Higher quality development

Dev

- Automate unit testing to verify that when code is created or changed it behaves as intended and that anything using that code will work properly as long as the unit test is passed. Tools like ABAP Unit can be used to develop and build unit tests within an SAP environment. Ideally, these tests should be executed automatically before transports are released, so the code can be verified before being moved anywhere.
- Make certain that your in-house, outsourced and contract SAP developers all work to the same code quality and standards by using a tool that ensures that:
  - All code is fully checked for coding standards, naming conventions and enhanced syntax checks
  - Potential performance issues are identified early in the development process
  - Any security issues are revealed to avoid unwanted breaches in production systems later.
- Introduce a peer review process where developers check and review each other's code. This is a great way to share knowledge, enforce quality and identify misunderstandings and mistakes before they leave development.



Outcome

The clear benefit of this approach is that nothing leaves development without being fully quality checked, minimizing costly downstream issues, errors and rework. In addition, the closer involvement of the business (via a product owner) ensures that customer requirements are clearly understood and the integration with the QA function establishes an early validation step into the development process.

Low-risk delivery for efficient testing

QA

A top quality development process will only provide significant benefits if changes can be successfully deployed to test, or any other downstream system, without risk. This can be extremely difficult in SAP environments, due to the highly integrated nature of the modules and the lack of effective change control.

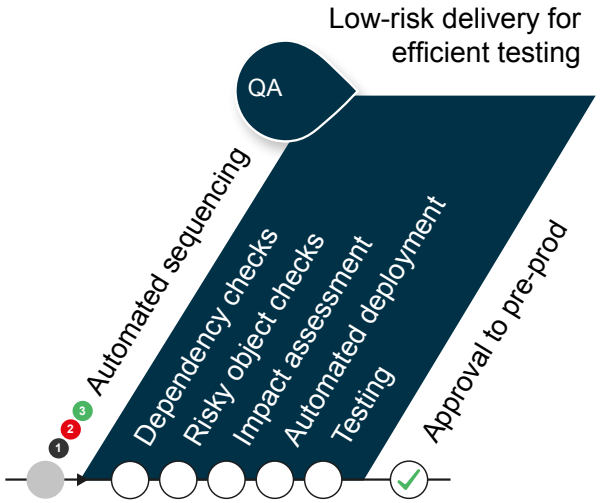
In many cases change and transport management is done via spreadsheets with no visibility of what is actually contained in them. This significantly reduces the speed of change, as it is extremely difficult, if not impossible, to understand how changes might relate to, and impact, each other.

What you need for DevOps for SAP

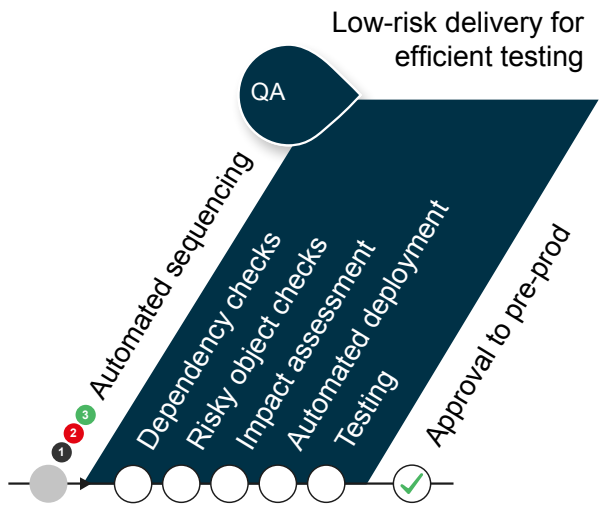
The ability to choose what you want to deploy, based on business priorities, is crucial to maintain a good cadence of delivery. An automation tool is a vital component of this continuous integration process, allowing you to avoid the delays and errors that manual processes can cause.

How you can do it

- Ensure inclusion of all SAP transports relating to the changes to be tested. A reliance on spreadsheets and e-mails to track all related transports increases the risk that some transports will be missed, resulting in deployment of a partial solution. This leads to issues during testing, wasted effort and needless rework. Effective tooling creates a traceable linkage between all transports and change requests.
- Ensure that transports are sequenced correctly so that different versions of the same objects are deployed in the right order. This will avoid:
  - Overtakes, where newer changes — for example an emergency bug fix — are moved in front of existing in-progress changes to the same objects. In this case there is a risk that a partial change, which isn't yet ready and so may cause problems, gets deployed along with the bug fix.
  - Regressions, where older changes overwrite newer ones that have been deployed out of order. Loss of the latest changes can result in removal of new functionality, or incorrect operation of parts of the system that rely on the most recent updates.



Low-risk delivery for efficient testing



- Make sure dependencies are managed carefully. Identifying conflicts between changes is one of the most crucial requirements for supporting selective deployment. It enables the effect of additions or removals from scope to be safely managed. It also allows a subset of changes to be deployed as soon as they are ready and any potential issues to be reported before doing so.

  Rigorous dependency management enables low level dependencies between changes, transports and objects to be revealed so that, for example, a program that references missing classes, methods, function modules, data dictionary objects, etc, can be highlighted to avoid deployment errors.

- Ensure that any objects that could cause harm in downstream systems are called out when approving or deploying changes. For example, SAP number range objects can cause major issues if not handled correctly. Consequently, transports containing these types of objects should be highlighted so that the appropriate action can be taken to avoid risk to systems. Implement a process to configure the list of risky objects and automatically report any transports that contain them, based on checks at various points in the process.

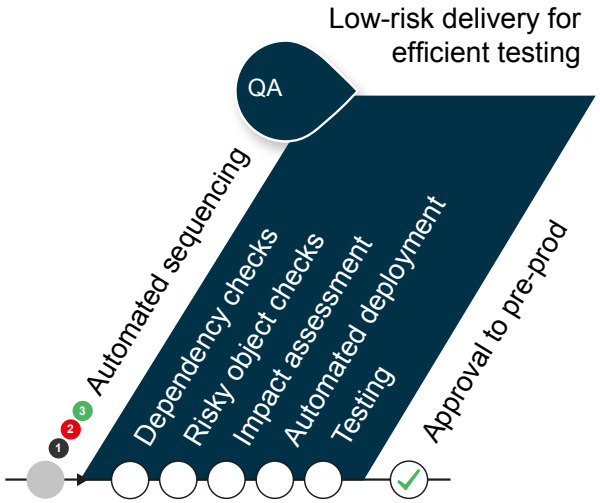
- Deploy an automated impact assessment process that enables you to understand the impact and effect that changes could have on business transactions and processes before deploying them into QA systems. Having visibility of the impact enables you to direct testing to specific business transactions or processes where it is most needed.

  The benefit is that the scope of testing can be more accurately assessed, so valuable test resources can be utilized more effectively. If you've configured the SAP Business Process Change Analyzer (BPCA) functionality it's also possible to analyze the impact of changes via that tool, to get a process impact of the changes contained in SAP transports.

Low-risk delivery for efficient testing

QA

- Automate the deployment of transports into QA and production systems after approval. This avoids a reliance on spreadsheets, emails and the manual intervention of the Basis team and ensures that only approved changes are imported at the correct scheduled intervals. However, it's important to consider the wider picture with regards to other connected SAP systems and activities, so think about how you'll manage and orchestrate cross-system deployments. An automated deployment process should ensure that dependencies across systems (e.g. between ECC and B/W or CRM) are automatically managed and sequenced, providing increased consistency and faster releases.
- Implement a method to track, notify and complete manual activities that can't be automated — for example, pre/post processing on BW changes or object activation in PO systems — which, if forgotten can cause huge inconsistencies and errors in your systems. These activities are often required before or after importing transports and are an important part of the change process.



Outcome

Ensuring that the changes selected for test are checked for completeness, sequencing, dependencies, risk and impact, removes many of the issues and errors encountered in delivering SAP change at pace. A more agile approach can then be adopted, where changes that are ready for test can be selectively and automatically deployed with confidence, rather than having to wait for everything in a bigger release to be completed. This accelerates delivery of functionality that the business needs and creates a much more responsive delivery model that can adapt to continuous change.

> ‘Elite performers’ employing DevOps practices deploy 46x more frequently than low performers, with 2,555x shorter lead times, 7x lower failure rates and 2,604x faster recovery times.

Accelerate: State of DevOps Report,
DORA 2018

Don’t break your current system

Pre-prod

When you’ve built and quality assessed all your great new features it’s really important to validate that your current systems won’t break when the features are deployed into production.

To do this you need to run regression tests to ensure that the most important parts of your system and processes are not adversely affected by the changes you want to deliver. For example, if you change something in finance you need to make sure the change doesn’t break your supply chain processes.

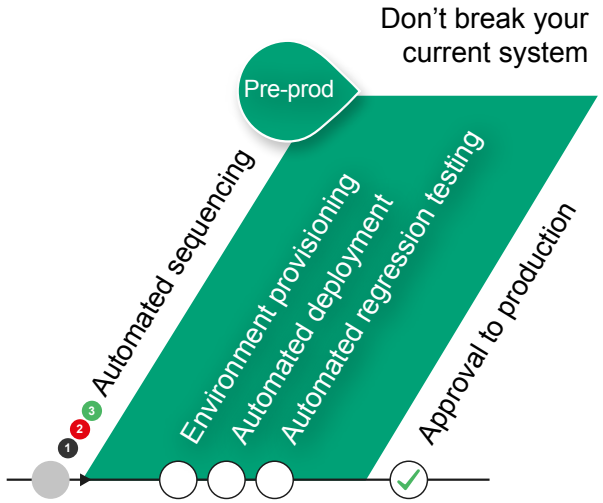
What you need for DevOps for SAP

A key requirement of continuous testing is to run regression tests on pre-production systems that are representative of the current production systems in daily use by the business.

Secondly, you need to ensure that your regression test process is robust, repeatable and has the necessary coverage to give confidence that the required changes can be deployed safely.

How you can do it

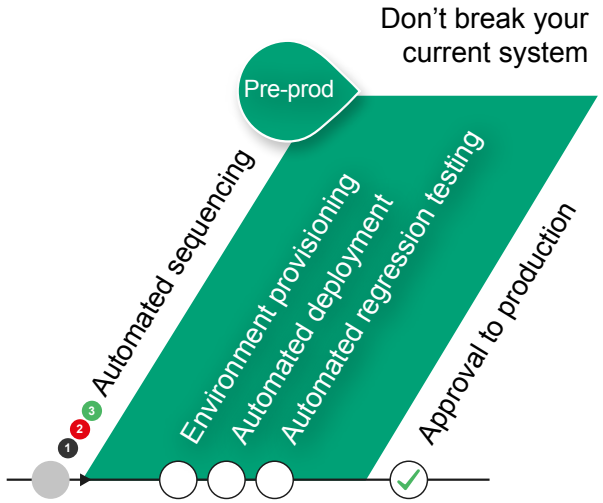
- Implement an environment provisioning process that enables you to quickly and regularly create production system copies, so that regression testing can be carried out on realistic and up-to-date data and configurations. Doing this regularly can be quite a challenge, particularly for large systems, but with the move from on-premise to cloud and virtualized systems becoming more commonplace, it’s a challenge that is becoming easier to overcome.
- Automate regression testing in order to remove one of the biggest bottlenecks of SAP delivery. It’s common for execution of regression testing to take weeks in a typical SAP environment. As a result, partial tests are often performed based on some sort of risk assessment of “what we think might be affected” — a very risky approach indeed.



Don’t break your current system

Pre-prod

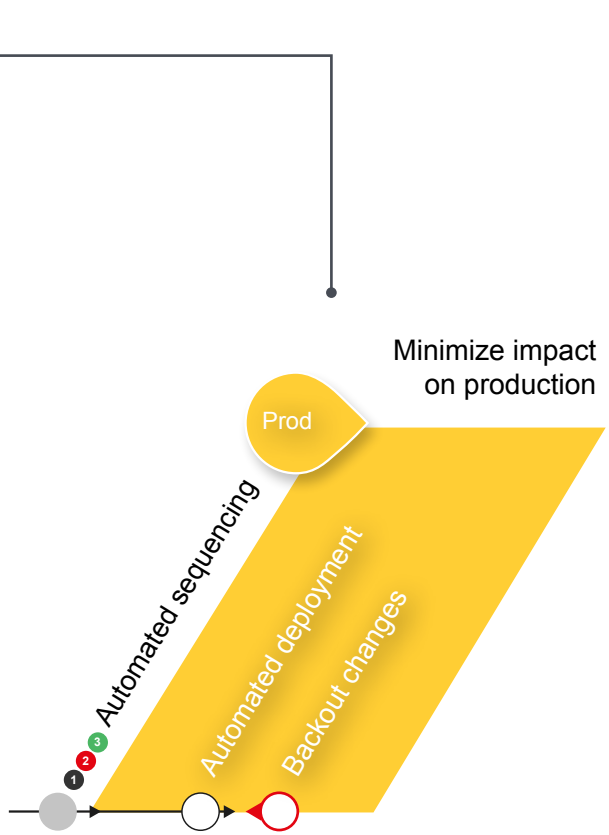
Automation provides the means to execute a full regression in as little as one day, so that resulting defects can be resolved in a timely manner before moving to production. However, challenges remain when using some automated testing solutions, not least deciding what level of coverage to include (i.e. balancing how much of the system you want to test, with what can reasonably be achieved) and keeping the automation scripts updated, as they will gradually become irrelevant as more and more changes occur.



Outcome

The key benefits of robust and regular regression testing are confidence and risk mitigation because any negative and/or unexpected side effects are minimized when changes are deployed to production. This translates into greater system stability and reduced downtime. Issues like lost revenue, reputation damage and lost productivity are significantly mitigated leading to happier customers, both internally and externally.

Reduce impact on production



The processes and controls mentioned in the previous sections will have a major positive impact on production deployments. They permit more regular change to become a practical reality thanks to a massive reduction in the risk of delivering change.

However, even the best development and delivery processes are imperfect. There is always a risk — however small — that an SAP deployment could stop the business by disrupting live system operations.

What you need for DevOps for SAP

Two aspects that are important in mitigating the risk of disruption when promoting changes to SAP production systems are:

- Controlled and automated implementation of the required changes, with any risks clearly highlighted up-front — critical for continuous deployment.
- A back out plan that acts as a safety net against business downtime by allowing you to expedite system and business recovery in the event of a serious outage.

How you can do it

- Implement the same checks and automation for completeness, sequencing, dependencies, risk and deployments when delivering change into production as you do when delivering change into test systems (see ‘Low-risk delivery for efficient testing’). These are at least as important, and arguably more so, when running production deployments.
- Introduce a plan of action to be taken that can wholly, or even partially, restore the system if a failure should occur following the deployment of changes to production. For example, if critical business transactions are negatively impacted (perhaps preventing sales orders from being processed), the ability to quickly restore the application to its previous state is vital to avoid significant business impact.

  Ideally this system restoration can be done “at the touch of a button” so that overall business impact is minimized. Traditional investigate-fix-test-deploy processes are often too slow and cumbersome, making it difficult to address serious problems quickly enough when they arise. This lack of a ‘safety net’ can be a significant barrier to the adoption of an agile approach.

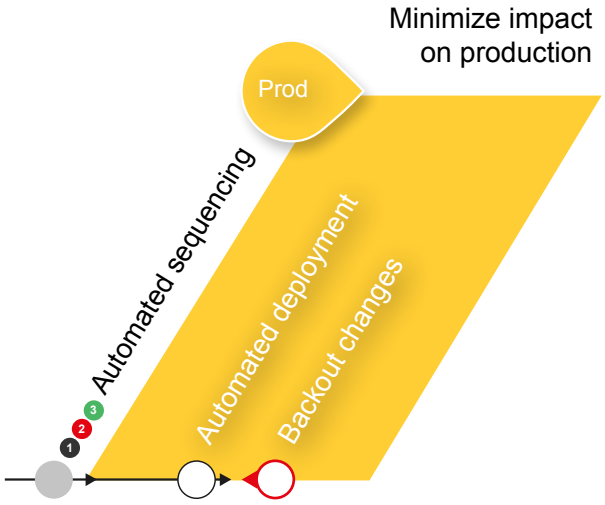
Reduce impact on production

Prod

Outcome

‘Stability is king’ has long been an unofficial mantra in many SAP environments. The inability to manage risk using traditional processes has proved a barrier to fast, frequent delivery of change (which in turn holds back the overall business). DevOps can maintain the required system stability whilst providing the means to deliver change far more often.

Rigorous controls to ensure that deployments are safely delivered to production will dramatically reduce the risk of system and business process outages, whilst an effective back-out plan will ensure that if disaster strikes, the impact of change-related downtime is minimized and business continuity can be maintained.



The human impact

When implemented correctly, DevOps fundamentally changes the traditional SAP development process. However, the manner in which DevOps impacts the roles and approach of staff can be just as important.

In DevOps effective collaboration is key. Traditional teams based on job function are replaced by multi-skilled, cross-functional teams who work together to deliver needed business outcomes.

The human impact

What you need for DevOps for SAP

Effective DevOps needs more than improved processes and a new organizational structure. Revised roles and responsibilities and a different, collaborative culture are also required. It is vital, therefore, for careful attention to be paid to definition of new roles and to securing the buy-in of all concerned, rather than imposing change in a top-down fashion.

How you can do it

- Gain management sponsorship. DevOps may struggle to succeed if it is initiated by the IT team alone. Managers need to create and promote a vision for the new ways of working, breaking down organizational silos, and removing the fear of failure. In other words, they need to make it attractive, safe, and rewarding for people to adapt.
- Make business stakeholders into product and/or process owners that take on the responsibility for steering teams towards the required business outcomes. Business stakeholders need to take an active role in every stage of the end-to-end development process, communicating, feeding back, and prioritizing where needed.
- Allow developers to take more responsibility for the quality of the applications they build by working with testing teams and adopting new methods like unit test automation.
- Encourage test teams to collaborate closely with developers from the start of the project and focus on user experience. They must also embrace test automation for functional and regression tests.
- Enable operations (Basis) to have early input into operational and infrastructure requirements. They must automate deployments and have a process for provisioning environments quickly to enable regular and frequent test cycles.
- Involve security professionals early in the development lifecycle so that security is built into the solution from day one, rather than added as an afterthought.

Outcome

Creating a successful DevOps culture empowers people at all stage of the development lifecycle to take responsibility for their part in the process, collaborate with their colleagues and focus on a common goal of rapidly delivering the high quality features and functionality the business needs to remain competitive.

Benefits and outcomes

In today's digital economy change happens fast and companies need to respond quickly. They need the flexibility to rapidly change, expand, extend and adapt their IT systems — including SAP — to drive increased efficiencies, seize new opportunities and maintain their competitiveness.

But, while accelerating the development and deployment of new features and functions is important in order to compete in today's dynamic business world, it cannot be done at the expense of business continuity. As reliance on customer-facing, revenue-generating applications grows, any downtime will result in lost revenue and opportunity and, potentially, damage to the brand.

Successfully adopting DevOps for SAP ensures that the 'engine room' powering those customer-facing systems can be updated and managed at a higher speed. With DevOps, speed, quality improvements and risk reduction go hand in hand so that companies have the ability and flexibility to change their SAP environments at the pace the business needs, confident that it can be done without compromising business continuity.

The end result is that SAP systems can be changed quickly, frequently, and safely. Starting today.

> "DevOps is a better way of working. We don't need any survival anxiety to show it is a better way of working. We know it reduces risk — delivery risk — and we know it increases quality."
>
> *Jonathan Smart, head of development services at* Barclays, July 2016

Key outcomes for our customers include:

Up to **70%** reduction in system outages and downtime

Up to **50** times faster delivery of SAP change

Up to **95%** less effort due to process automation

Conclusion

DevOps is a powerful enabling force for driving business change. First class automation tools that support the entire development and delivery process, from development to production, are an essential part of this approach.

Basis Technologies DevOps and test automation platform is built around the core concepts of automation, safety, quality, visibility and audit to put organizations in complete control of their SAP change and release processes. The products support ALM, agile and DevOps approaches, allowing organizations to transition from waterfall development to continuous delivery at a pace that suits them.

> Digital business demands a high degree of automation for improved speed and effectiveness of software delivery. Product management leaders should focus on tool functionality that supports agile methodologies, automation and collaborative relationships to meet the challenges of the DevOps goal.

Market Trends: DevOps — Not a Market, but a Tool-Centric Philosophy That Supports a Continuous Delivery Value Chain.
Laurie F. Wurster, David Paul Williams, Thomas E. Murphy; Gartner, October 2016

The DevOps and Test Automation Platform

ActiveControl allows SAP systems to respond to new business requirements at high speed. Its range of powerful automation features support agile development, DevOps and Continuous Delivery in SAP environments, generating more business value through faster, safer application delivery. It accelerates change without putting business continuity at risk, providing fast, efficient change management that puts you in complete control of your SAP change and release processes.

Testimony is a fully automated, next-generation testing tool which uses Robotic Test Automation to create a comprehensive regression test library and eliminates the need for test script creation and maintenance. Without scripting, regular regression testing becomes a reality in the short ‘sprint’ (development) cycles typical of agile and DevOps approaches, and provides a practical way for development teams to shift testing left.

At Basis Technologies, we develop automation technology that massively reduces the time and effort needed to execute SAP change and testing. We are committed to driving business agility and transformation through agile development, DevOps and continuous delivery.

Our software — the only complete DevOps and testing platform engineered specifically for SAP — enables companies to accelerate innovation, ensure continuous quality and delivery, and lower risk across even the most complex SAP landscapes.

Since 1997, we have helped enterprises become more agile, innovative, and competitive. Our platform is SAP Certified for use on both S/4HANA and ECC systems, which is why many of the world's leading companies trust our subscription software to help them succeed in the digital age.

Contact us to find out how we can help your business adopt DevOps for SAP

www.basistechnologies.com