DynamicIQ

Zero Failure

The Future of Application Configuration is Automated

White Paper

Executive Summary

During the past decade, businesses have embraced smarter business tools, cloud solutions, specialist IT services and new ways of working. This has made companies dependent on data integrity and application availability.

The result? Wildly different application ecosystems. If you are involved in any part of software development, the above should sound familiar. You will also know there are plenty of best practice resources available, however the complexity of application configuration and the deployment lifecycle means that your team likely has its own set of challenges.

This is because behind every application lies a unique blend of people, technologies and policies. When it comes to application deployment, some organisations might invest in internal Production, Test and Quality Assurance teams, while others leave application configuration and change management to their software integrator or application provider.

This shift in responsibility and accountability offers its own challenges. Application providers sell solutions on the promise that their technology will scale and perform 24/7. Therefore, from their perspective, every configuration change must be managed efficiently, deployed effectively and analysed quickly. Downtime results in reduced profitability and dissatisfied customers.

This white paper discusses the current state of application configuration and the deployment cycle. It addresses the importance of maintaining a reliable configuration pipeline and the options available for teams searching for methods to streamline operational management while reducing exposure to risk and non-compliance.

By analysing the benefits possible from automation, readers should come away from this white paper in a stronger position and with a better defined configuration strategy.

The Current State of Development

Our first step is to define the role of application deployment within the context of wider software development.

Development is never in isolation; it is a continual process where existing code is refined to deliver the desired business outcomes. Software development is an 'iteractive lifecycle'.

This means its underlying processes are regularly repeated when new functionality is demanded, technical issues identified and external factors change.

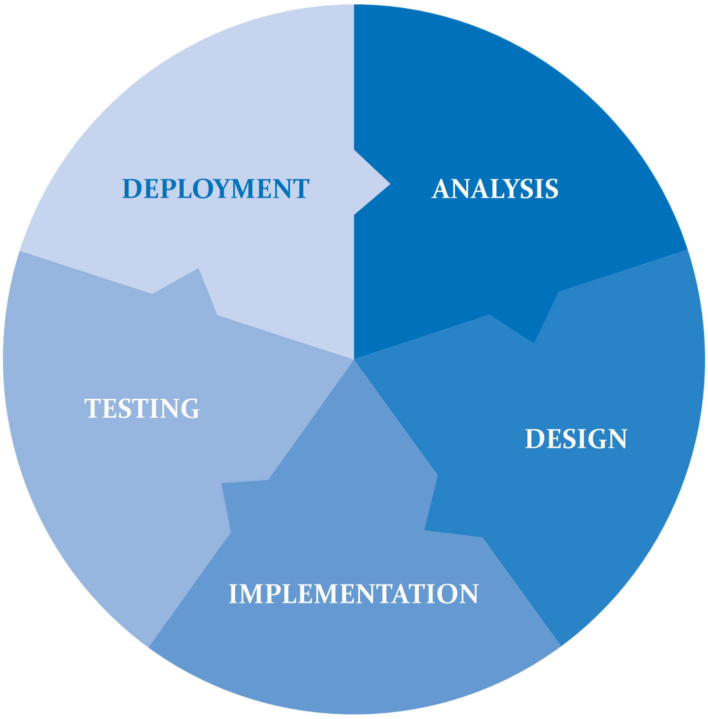
There are countless different development lifecycles, however the process is commonly segmented into five stages – analysis, design, implementation, test, and deployment.

Within these stages are three key issues that can have a negative effect on configuration:

1. **Inconsistent tools and processes**
2. **Over-dependence on manual working methods**
3. **Inconsistent approach across application environments**

All the above contribute to production instability. Unfortunately, as is often the case with highly distributed application environments, there is no silver bullet to solve these challenges.

There are, however, two streams that drive tangible improvements; one strategic, the other tactical.



A typical development lifecycle

A strategic approach generally involves defining, documenting and implementing common standards, tools and processes. This takes time, effort and resources, but in the long-term delivers impressive results. Tactical improvements include activities to solve technical issues or smaller scale changes that deliver immediate benefits.

This is where automating configuration shows its business and operational value.

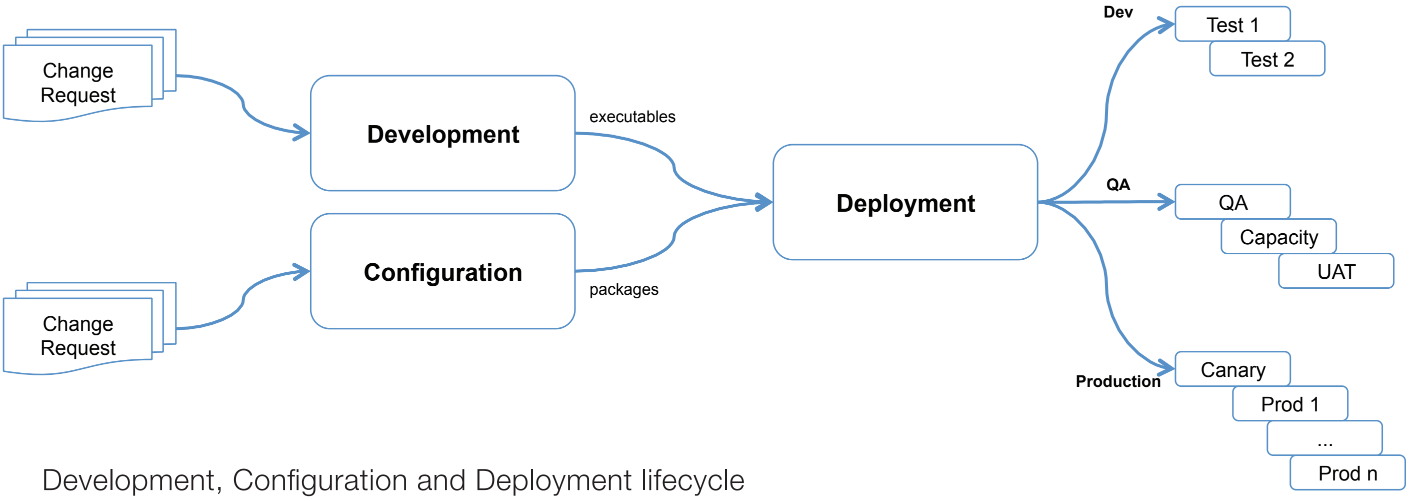
Configuration's Role in Deployment

Irrespective of whether changes are strategic or tactical, most configuration processes will follow the same structure.

From a technical standpoint, once a new feature, bug-fix or enhancement has been agreed, alterations will be made to configuration (packages), code (new binaries) or a combination of both.

This takes place in files found within environment instances (a collection of files that deliver a specific function). These files are interdependent, assigned to a specific instance or shared across numerous distinct configurations.

The scale becomes even larger once 'application properties' are involved. Each configuration file can consist of hundreds or thousands of properties. Certain properties are consistent across every instance; others are intentionally diverse for functionality reasons.



Development, Configuration and Deployment lifecycle

As expected, someone or something has to generate these files and binaries. There are various options available to businesses:

- ✓ **Manual –** editors work through files making changes one-by-one. This is incredibly time-consuming, inefficient and error prone.
- ✓ **Semi-automated –** generating files using templates is possible when basic configuration is involved, but not for highly structured applications or distributed systems.
- ✓ **Full Automation –** more advanced solutions that offer the ability to replicate changes across extremely complex application environments, irrespective of data volatility or the pace of change management. These platforms have only come to market in recent years.

Each option offers distinct benefits, however the growing acceptance of DevOps means development teams, IT departments, software integrators and operational staff are considering the workflow advantages of a fully automated solution.

After all, as applications grow in scale, trusting such a crucial element of software deployment to people alone is too much. The risks that accompany application failure are simply too damaging to leave to chance.

> “By 2020, DevOps will become a mainstream strategy employed by over 25% of Global 2000 organisations.
>
> Source: **Gartner, 2016**”

Regulation, Compliance & Preventing Mistakes

Angry customers, unproductive employees, regulator pressure and financial penalties. Application reliability is undergoing a huge degree of scrutiny.

Each of these scenarios is associated with application failure. Software errors can be even more detrimental to an organisation when the sector is highly regulated. Reduced profitability, reputational impact, and shareholders running for cover are three common outcomes.

Financial services is one industry experienced with these risks. Application availability is fundamental to long-term success. Regulatory penalties extend into the millions of pounds, particularly since the introduction of MiFID II and other regulations that affect data and software governance.

Due diligence is imperative. This statement holds even greater weight once non-technical processes are included in the equation. Every configuration change needs approving, implementing, testing, recording for compliance and then further quality assurance before deployment.

Humans alone cannot cope, and time-pressures only increase the chance of human error.

Applications and new features must be launched in weeks, days or even intraday. Waiting months for a new feature is no longer a viable business model.

A new approach is required urgently if teams are to keep up with the expectations of the business.

> “MiFID II is a wide-ranging piece of legislation and, depending on your business model, could affect a wide range of your firm’s functions – from trading, transaction reporting and client services to IT and HR systems.”

Automation Meets Configuration Management

So, how can developers and IT departments prevent these nightmare scenarios from occurring?

Invoke the power of technology. New automation tools remove the need for manual processes, increase the pace of change management and ensure a more predictable configuration pipeline.

Complete accountability and authentication is attainable. Individuals no longer have to record every single change to configuration. Instead, tools such as Application Modeller record who made a change, when it was made and the parties involved in the process.

This strengthens compliance and auditing and enables managers to quickly identify what is wrong before an application ever makes it to deployment.

Automation does require a new way of working. Some application developers might be apprehensive about trusting such an integral part of the development process to technology.

But, look at the evidence in the included case study – zero application failures. This is just one example.

In the world of international finance and fast-paced software development, zero failure is the level of reliability every organisation strives for. It is what regulators expect.

With more and more automated solutions emerging for configuration management, it is important to assess current processes and consider whether existing software solutions and workflows are holding up.

Ullink is a financial services software provider which is trusted by 150 of the world’s leading banks and brokerage firms.

The company has been using Application Modeller to automate application configuration.

On one project over 18 months, there were 85 release changes with zero failures.

Evaluating for the Future

To assist with this process here are key questions to evaluate current progress and to identify next steps. Ask yourself:

- ✓ How often do failures take place?
- ✓ Can we quantify time savings?
- ✓ Has there been any reduction in production incidents?
- ✓ How long to reach deployment? Is this faster than six or twelve months ago?
- ✓ What percentage of our operational and IT teams are still involved in configuration?
- ✓ Do we possess real-time feedback and visibility into configuration changes?
- ✓ Are current solutions delivering a tangible return on investment?
- ✓ What risks remain?

Answering these questions confidently should be the first step towards better application configuration. Any question without an answer will identify which operational and technical areas require more attention.

The questions also operate as a baseline for measuring results once automation begins, or as a means to build a strong business case for implementing a solution, such as Application Modeller.

Ultimately, whatever stage of the automation process your business is at, a clear plan of action is needed. Without one, improved authentication, authorisation and accountability is nothing more than an unobtainable goal.

These outcomes are no longer out of reach. They are achievable by every business and software team. You just need to know where to start on the road to automation. Ideally, this white paper has revealed the appropriate path to the results you are searching for.

About DynamicIQ

DynamicIQ is a fast growing software company headquartered in London. Established in 2013, the company designs solutions that deliver tangible business improvements and drive productivity.

Each DynamicIQ founder has over 20 years' experience in the highly regulated sector of financial services and has spent his professional life immersed in simplifying the complexities associated with DevOps and large scale application development.

The company's software solution, Application Modeller, has everything an organisation needs to automate application configuration. Application Modeller is the simplest yet most accurate way to manage configuration changes with minimal risk.

Application Modeller has quickly become known as a powerful solution for development teams, operational staff, IT departments and software integrators.

Contact Us

✉	hello@dynamiciq.com
🖥	www.dynamiciq.com