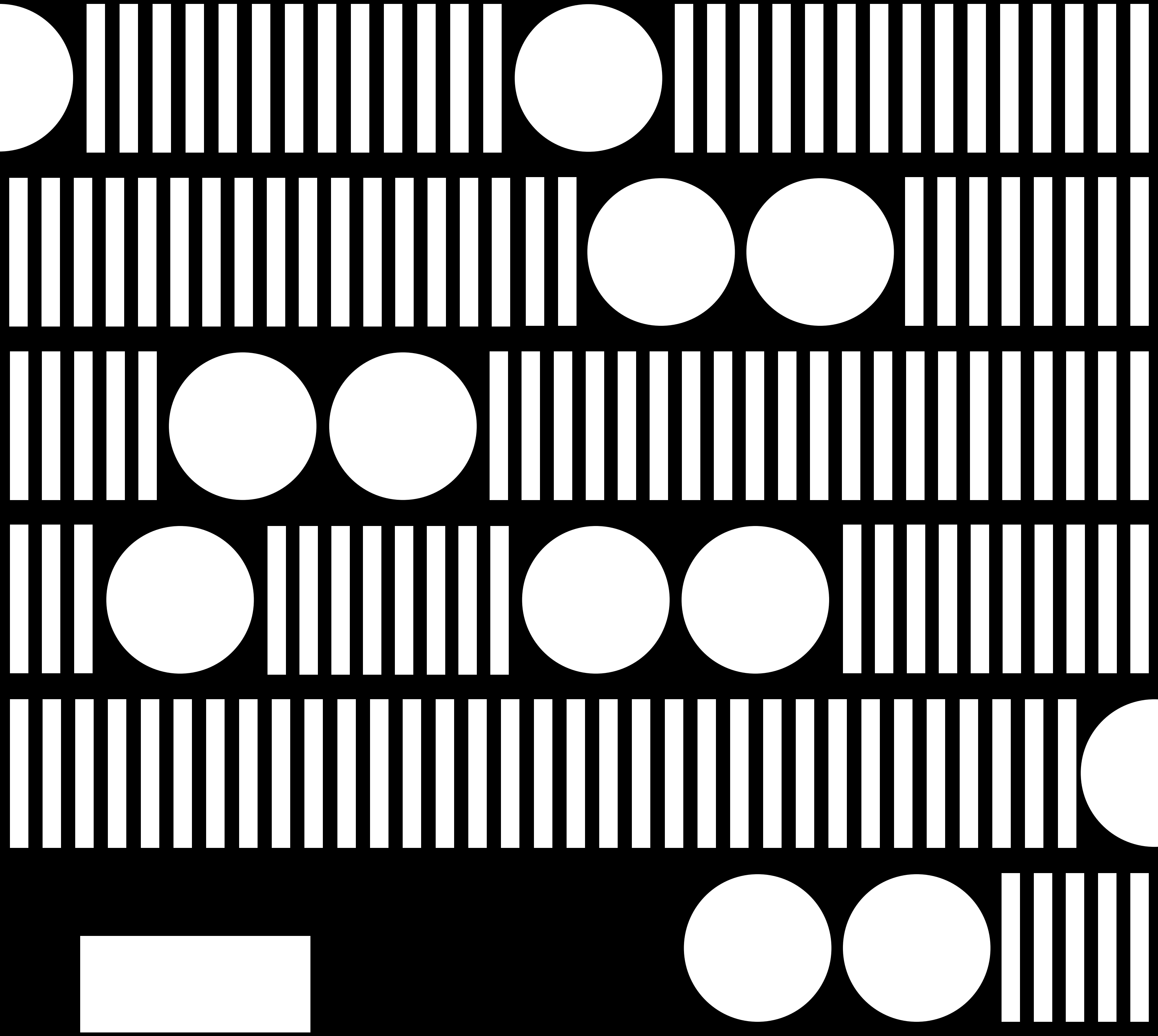
WHITE PAPER

No.00

The Future Of Continuous Testing

Author - James Briers



Contents

Introduction

IDS are recognized experts in freeing software and data projects from bloated, outdated quality practices for digital business assurance. Our focus is to educate teams, speed up processes, reduce costs and increase confidence in quality engineering for software and data.

We help influential leaders in software and data quality all over the world. We provide cutting-edge tools and strategies for solving real and meaningful quality inefficiencies with the aim of delivering good quality habits and making teams and businesses more successful.

This paper discusses what the real issues and inefficiencies are in delivering quality and walks a reader through real solutions which help lift the mist on the mechanics involved.

Executive Summary

Teamwork, discipline, good habits, structure, roles and responsibilities are the DNA of any successful team.

Let us look at sports teams as an example. They are made of people with different strengths and so are better suited to different activities which suit their own positions.

The most successful sports teams have specialists in each position. Those that don't recruit specialists are mediocre, mid-table performers. Utility players have their place. These are vital when you have your structure in place and unforeseen circumstances mean your specialists are not available.

Leading and coaching successful sports teams relies on giving players & athletes a clear game plan or set of instructions to follow. Giving a person a role in the team that complements their own personal skillset will reap great rewards for the collective group. Having a motivated and happy environment for players to work in also benefits greatly. If these factors are achieved, everyone as a team will achieve the outcome that they are all striving for.

What does this have to do with software and quality? The two are very similar in structure, with people, process, skills and tools used. The only difference is that our opponents are the challenges we face along the way, those road bumps keep us focussed and those that aren't foreseen will just force us to adapt our strategy as they occur.

Having the right team strategy will deliver success. Have people deliver their specialism and stay in their lane. Don't have manual testers toying with automation. Generate good habits within the team and ensure that talent is supported by good structure and direction.

Our robust, proven, purpose-built data migration and assurance process takes care of your data at each step in the journey, ready for use in your organization's internal and bespoke systems.

You need specialists to design, implement and deliver the winning system. Once implemented, through coaching and mentoring, your B-Players will become A-Players, when working inside a robust framework with a clear strategy.

With all the above in mind, having structure and strategy is only part of the game plan though. We need to measure our outcomes and display progress. In sports, we have a scoreboard to present up to date and on-demand game status and a clear, winner or loser.

How do we understand how well our project is performing? In software and data projects, how are we presenting our results? How do we keep pace with ever-changing metrics in the modern world of software delivery? project dashboards are handcrafted in an an application such as Excel or Lucid, or handled using a project management system. However, they still typically require manual intervention on a regular basis.

Project automation eco-systems; are they nirvana, or are they more achievable than you think? In this paper we explore what is required in order to provide the correct quality-focused on-demand metrics.

Our focus is to implement a quality engineering solution which not only assures business digital change, but provides up-to-date, on-demand views of the project status.

All dashboards are linked back to your project or programme contract through to your final release note and all artefacts in between.

Our focus is to repeatedly deliver quality engineering with intelligent automation, digital business assurance and fully automated dashboards to provide a single version of the truth for all programs and projects.

What is the Problem?

Busy projects and bad habits lead to inaccurate reporting and dashboards.

Advances in technology and innovative delivery methodologies have resulted in rapid software development and deployment practices such as Continuous Deployment or DevOps. The speed at which software and data are delivered is growing on an impressive scale.

Having fully automated processes for code deployment and environment management is great until you come to measure quality and progress being made. Hand-cranked reports are costly and inefficient, and creating dashboards from manually generated metrics is even more problematic.

Weekly updates and meetings become dull and resented, resulting in a severe loss of enthusiasm from the management team.

Businesses enter projects and programs to propel the business forward. This should be an exciting time for all involved, and it usually is exciting until we hit a status quo.

Being part of something which is new and in the creation stage is brilliant. It is fresh, new and challenging getting the processes set up, designing the frameworks and approaches. However, the ‘norm’ sets in when the project is inflight and those tasks which are outside of your development or test plan start to bite.

Shortcuts are taken, and a certain amount of manipulation is required to get that report out. Bad habits occur and this is when mistakes happen.

> **“Industry research indicates that automated testing levels consistently hover around 20% from a coverage perspective.**
>
> Source: Gartner

Reports and Traceability

All projects are connected to a contract, regardless of whether an external party is delivering, or the project is delivered by an internal team. Stakeholders want - and have a right to know that - what is being delivered is what they are investing in.

If reports and dashboards are being hand-crafted, then what is the true status?

Remember, each discipline within a project carries their own project artefacts. It takes an immensely disciplined team to ensure that all artefacts are linked back to the requirements correctly. This process becomes even more difficult when it has to be retrofitted into thousands of related artefacts. Once again we are pinched by a manual fix to a complex problem.

Without full traceability from contract to project assets, it is near impossible to justify the position of delivery at go-live and prove that what has been delivered is what had been agreed at the start.

From a contract, you work forward and Epics, User Stories, Requirements etc. must all be linked through to lower-level project artefacts. Often this is not the case, and artefacts become orphaned or linked incorrectly to the wrong parent artefact.

Poor habits around traceability management are compounded when we are looking at status reports

Development and Quality Assurance Artifacts

Projects are busy and stressful when they are in-flight, and in a perverse way, this makes them fun and exciting. It is easy to get carried away with the cool and creative side of the project, let bad habits creep in and stop performing simple tasks which are perceived to have no benefit.

Lack of artefact traceability is one of the biggest contributory factors to project delays. The reason for this is because project teams are unclear of exactly where they are in the timeline due to inaccurate reporting and dashboards.

This program overran by 18 months and cost the organization millions of additional pounds to get it on-track and complete. In the modern world, it is imperative that we are all contributing efficiently and performing good habits to prevent such overruns and delays.

Rapid Deployment

Development and delivery practices are evolving and becoming increasingly efficient, resulting in the ability to implement newly developed code changes pretty much every minute. How are we monitoring the process around this? How are the results of a unit test recorded? Are bugs found by unit tests recorded and managed with a correct process?

With speed and agility, the artefacts are created, yet they do not go anywhere other than a development file.

Build and deployment process documents will exist in 90% of cases. However, in terms of this being a traceable artefact, the complete opposite is true.

> “
>
> **More than 80% of projects run over time and/or over budget. Cost overruns average 30%. Time overruns average 40%.**
>
> - Bloor Group Alignment

From a deployment perspective, not linking the build and deployment plan will have a knock-on effect on other artefacts such as release notes, unit test results, functional test results and bugs. Often all these outputs from an efficient approach to deploying code are not linked to a requirement, user story or release.

DevOps models are put under significant strain if the dev side of the bargain fails to evidence that it has a high level of quality attached to the most current build and deployment of the code being shipped into operations for go-live.

So not only are the business unclear on the status of the quality levels, the fellow tech teams who are owning the new release have no real confidence in the code either due to lack of visibility or worse, lack of confidence in the reports being generated.

Test Execution Results Uploading

An area often massively over-looked is the process of marking test execution results. Test results are often recorded correctly when executed manually. However, some automated test solutions struggle to map execution results back into the repository.

Automated test results are then manually updated into the repository, a process which can suffer from human error when mass updates are performed.

Following automated test execution, this manual update becomes a natural bottleneck and especially so when development is deploying new code more regularly. Test execution turnaround at such a cadence becomes extremely difficult to keep up.

Manual Processes Are not Always a Bad Thing

This process is not just about a pass/fail update, bug management is another area which requires manual intervention and rightly so. This is a process which requires meticulous attention to ensure that time is not wasted on having illegitimate bugs sitting in the developer’s queue.

Manual validation of bugs, through test execution, is prescribed to establish the repeatability and legitimacy of a bug. The bug then should follow the internal triage process, be fully evidenced and have traceability to the test, which in turn can be traceable back through all other appropriate assets back to requirements or user stories, epics and even the program contract.

So not only are test assets not fully traceable back to requirements and other project assets, it is highly likely that those results are inaccurate.

> “
>
> **We have seen test outputs feeding in overarching management reports and failing to show a correct picture of what has happened.**

Quality assurance teams have declared they have manually executed thousands of tests in a release, yet the reports shared with senior stakeholders are misleading and showing a more positive picture.

We recently observed a program which was two weeks from user-acceptance-testing phase (UAT), which was to be immediately followed by go live. The stakeholders believed the program was ready to exit the systems integration testing phase (SIT), having reviewed a series of highly favorable status reports.

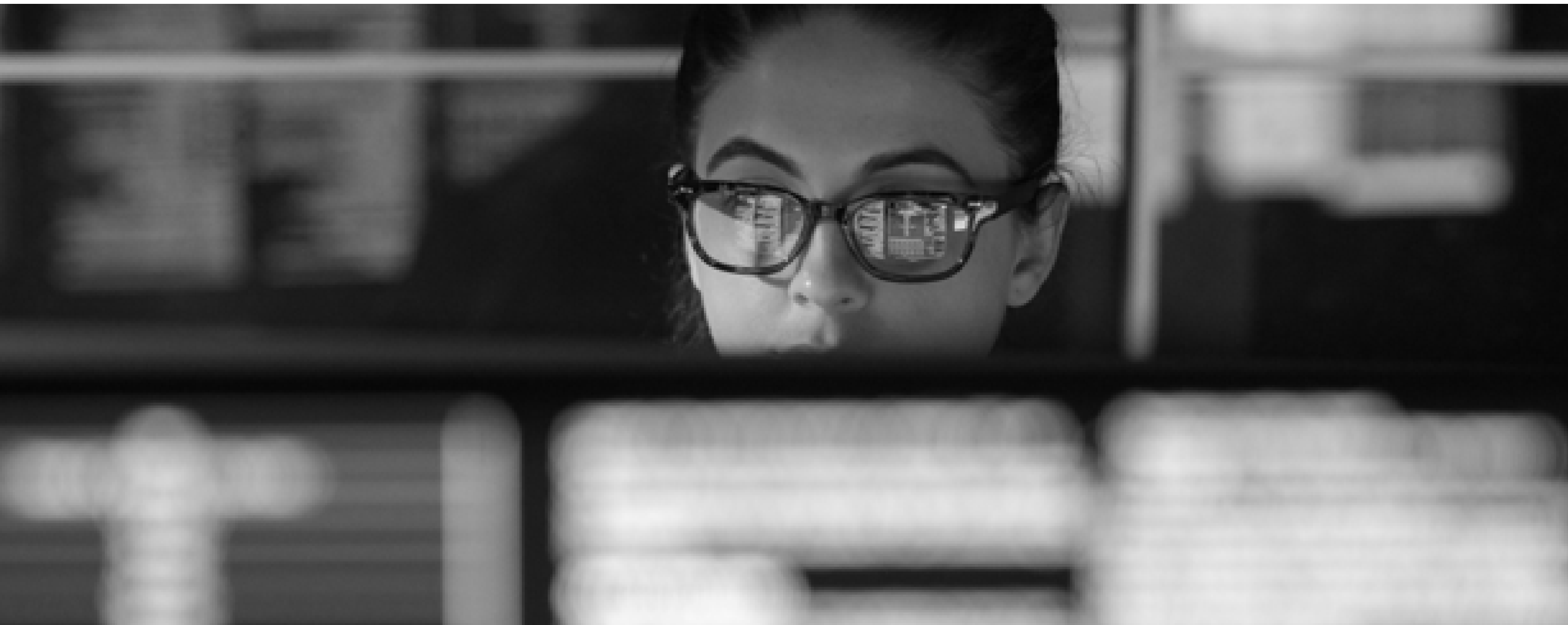
When delving deeper, the reality of the situation was that out of 8,000 tests, 2,500 tests were in a non-executed status. This is an alarming statistic. More than thirty percent of the tests had not been executed, meaning that there were no results available to allow program managers to determine the quality status, or application functionality.

Following days of investigation, it became apparent that while testing had executed some of the test cases, over 40% of the test cases had no link to any other artefact in the project repository and had been completely orphaned. When reports were being generated, it showed that a higher percentage of the testing had been completed against the delivered requirements for that release.

To further compound this, it appeared that the business analytics and development artefacts were in a similar state.

The team had to quickly refactor the reports to bring them in line. This distracted from any delivery focused activity for more than two weeks, with a team of 20 people. 200-person days were required to bring the reporting back in line.

This, unfortunately, is not where the story ends. Had the team corrected the lack of traceability links and enforced a suitable process going forward, they wouldn't have found themselves in a similar position three months later.



Checklist

Is your organization currently experiencing issues around program or project metrics? If so, let’s look at how you take it to the next level.

Challenge	Have this issue? Y/N
Have you invested in automation, but struggle to benefit from those expected efficiency gains?	
Are the metrics out of test constantly under scrutiny?	
Does your automation require a lengthy implementation period before you can benefit from executing your tests?	
Is your existing investment in automation now proving too costly to maintain?	
Is your BI-testing strategy is failing to deliver the right level of data for stakeholders?	
Do you have a perceived lack of time, budgets, and desire to invest in automated testing from stakeholders?	
Is there a lack of confidence in the opinion of the tester?	
Is your test team predominantly made of manual skill sets?	
Do you have a young and in-experienced test team lacking the confidence to try new techniques of testing?	
Has a lack of planning and insight led to failure by focussing on low-risk areas to automate testing?	
You are struggling with successful automated testing due to a lack of understanding of how an automated testing ecosystem is designed?	
Are your test data and environments causing issues with consistent test executions?	

Why Hasn’t the Problem Been Solved?

Having the right knowledge and controlled solutions is challenging and demands consistent tenacity from organizations.

To implement this level of control is difficult and tricky to Navigate. It requires knowledge and understanding of how traceability works and how reporting is required from a stakeholder perspective.

Knowing Your Destination is Important to the Decision-Making Process.

Understand your project's artefacts and how they are processed. It requires absolute conviction to design, lead and implement a winning eco-system, and this is further compounded when trying to fix a project or program in flight.

As an ambassador of quality and leader of the project, implementing a solution is tough and requires a level of tenacity beyond what can be taught. It has to be within you. Bad habits and cutting corners are the biggest areas of concern. The process to manage artefacts is something which can be documented, taught, and implemented. Enforcing good habits can be extremely difficult especially if the team is used to doing things their own way.

Implementing and enforcing a traceability solution is extremely time-consuming and requires a dedicated person or team to evangelize this approach. Investigating the current state is an involved process which requires workshops and team engagement, before using the data extracted from the team to design the solution.

This is especially disruptive in the case when a project is in flight already. Distracting team members to focus on this initiative will cause expensive delays to the project.

You Only Know What You Know

If you have not experienced working within processes and frameworks in previous roles, it is difficult to demonstrate credibility when trying to implement sophisticated and automated processes. When implementing traceability processes introducing tools and techniques unknown to other team members, it is essential that this is led based on credibility to handle objections with real experiences.

Being time-poor allows for corners to be cut and any changes and good processes are quickly disregarded for shortcuts or bad habits. This will only land you with greater delays to the program in the long run.

Often, great ideas are floated, peers dismiss too quickly and go back to a manual first approach.

Understanding the plan and outcome requirements help focus the business digital assurance and quality engineering on focus too. Knowing what to test with reporting in mind is difficult and easy to get wrong, resulting in over-testing or delivering the wrong type of testing at key points of the project.

Pre-Planning Cannot be Ignored

Lack of faith in automation is one of the key factors behind the traceability and automated dashboard problem. It has been tried by many with limited success, leading to a retained belief that manual traceability process are a requirement. Failure to understand the desired outcomes, and a lack of understanding of tooling capabilities are one of the leading causes of such failed attempts at automation.

Pre-planning is vital. It is essential that the outcome is defined with stakeholders, understanding what is required when reporting quality status to support decision-making.

This process is tricky, as stakeholders often are not clear on what they require until deeper into the program.

To plan the implementation of clear and informative automated dashboards takes effort. An understanding of the bigger picture coupled with an ability to break down the deliverables is key. Selecting team members with skills to implement and govern the process is required. However, this is not as easy as it sounds, especially when the team includes member who are complete novices in the tooling.

Building out the framework is the key to success, yet it is also the doorway to failure. The team implementing the plan is key and, using specialist resources is an essential requirement. However, this brings in a further complication as having a team of A-Players is costly especially for lengthy project support.

What Could the Future Look Like Without Manual Errors?

The problem is that teams are not spending enough time collaborating, failing to grasp concepts.

Align business goals to project requirements, link project artefacts to contractual outcomes and expedite fund release cycles for your programs!

Business and technical teams spend too little time collaborating, failing to fully grasp benefits and challenges around what the outcomes of their goals mean for the business. This compounds the issue when delivering the tasks required to deliver those goals and a natural disconnect appears between business and technical goals.

We start by making decisions at a portfolio and program level to have a clear initiative on the presentation of metrics. This helps steer decisions associated with releasing the technical solution which ultimately supports the business goal.

The metrics will be converted into a reporting and dashboard structure that provides an on-demand view of the program of works. With the metrics in mind, the program ensures that the processes, designs, artefacts and documentation are generated in-line with the business and technical goals.

Implementing lean principles to automate processes where possible, will speed up and improve quality and ensure traceability is in the DNA of the program.

IDS have a combined Test PMO & Governance with automated continuous deployment quality assurance services that, when used together, will accelerate quality control and the management of the assets and data associated.

This grouping of solutions presents an end to end, fully traceable, demonstrable and on-demand set of dashboards which provide a lens into project status and quality in real-time.

Explain the traceability starting from epics or high-level requirements every child artefact is generated from within its parent.

We have an opportunity to bridge the gap and reconnect the business requirement with technical goals by ensuring a close relationship between what is the desired business outcome and the technical interpretation and implementation of how that goal is realized.

Example

The design phases will take place between Business Analyst, Developer and Tester in each sprint, and will be used as an opportunity to add more detail to the high-level user story and information provided by the Business Analyst.

- Epic explained
- Business Analyst to create and establish user stories based on business requirements, from within the associated epic
- Business Analyst to define initial acceptance criteria for each user story
- Business Analyst to set up User Story Analysis workshop on the final Friday of each sprint
- Developer to describe the potential solution and document within the user story
- All development artefacts are associated with the user story including release notes, build and deployment esults
- All provide input to define acceptance criteria
- When the solution is in "in development" testers should prepare an Acceptance Test(s) and document as part of the user story
- All test scripts are created in the test repository and linked back to the associated user story or stories
- When a user story lands in "Ready for Test" on the task board, begin setting up your test execution if manual, or execute tests using automation as part of the daily/ nightly build
- Record your testing evidence against test scripts, which should be now associated with the user story
- Report any issues to the developer, raise valid bugs in test solution which automatically updates into Jira, and associate them to the user story automatically.

Using the very high-level example above; all data is entered in real-time as it happens.

When code is deployed; all tests are executed including unit tests, system, and system integration tests. Once the execution is completed, all results are uploaded automatically into the test repository. They are then mapped against the associated manual test scripts.

At this point, all reports are reflective of the latest quality status of build results, from a development and test perspective.

All defects are validated manually, and legitimate defects are added to the repository and linked to the failed test. Any tests which failed and are illegitimate failures, are re-executed and the results are updated once completed.

Once again, the reports and dashboards are shown in real-time.

A stakeholder can be on their way to the office on a train and want to understand how the overnight test execution went. They open their phone and look at the configured app to view the dashboards in preparation for early morning updates.

What is Different About our Approach?

Our concentrated methodology ensures we are delivering excellence while accelerating automated quality engineering.

IDS accelerates automated quality engineering for enterprise application testing on SAP, Salesforce, ServiceNow, Oracle, and many other popular enterprise applications This is so your organization can innovate faster while reducing business risk.

IDS’ Kovenant methodology and approach follows a concentrated, set path to ensure we are delivering great results and delivering excellence within the framework.

1. Identifying the business and technical problem
2. Define the solution
3. Design and implement
4. Test and validate
5. Continuously measure and repeat.

We work within a set repeatable framework. IDS’ approach builds and implements a solution based on facts derived from the health-check.
The solution is implemented, measured, and constantly sampled. This ensures everything is still working to support a continuous opportunity for improving and evolving the original solution.

Good habits are easily implemented, they are more difficult to keep up. Teams must have processes and habits re-programmed in the new world over a series of weeks and months to enforce and reinforce the new habits supporting successful program outcomes. This is what our process does.

Health Check

Every journey begins with a single step. This is our first step in understanding more about your problems. IDS focus on using Jira as a project solution due to the ability to integrate with many other repositories in the marketplace which have demonstrated many successes including on sites which have Azure DevOps in place.

At this stage, analysts will have an understanding of the pains you are having as a program which helps understand the direction to go in.
For a solution to be understood and narrated with a solution, it is vital that a full deep dive into your current state is delivered.

The focus and output from the health check provides insights into the following areas;

- → User management
- → Workflows
- → Issue management
- → Defect management
- → Projects
- → Security settings and policies
- → Audit logging
- → Project roles
- → Global permissions
- → Issue collectors.

What is Different About IDS's Approach?

- → User interface
- → Import & Export
- → Mail
- → Admin helper
- → Shared items
- → Automation
- → Advanced traceability
- → Test management configuration (Zephyr/qTest)
- → Test management custom fields (Zephyr/qTest).

Recommendations within the report provide a high-level set of tasks to focus attention towards to help drive the implementation and solution plan design. Along with the recommendations, how the new world will look for the organization is depicted with a series of clearly defined dashboards.

IDS Project, Sprint and Quality Management

Starting at the beginning of the journey with contract traceability management, IDS ensures all paths lead back to the supporting reason for the project existence. This protects the stakeholder, vendor and any customers who are involved in the program.

Establishing the links to and from test artefacts such as requirements, test cases, test results, bug management processes, back to high-level requirements or user stories which are all linked to business requirements. Ultimately the contract ensures all other business intelligence are delivered accurately.

It is vital at this point to set up supporting test management functions with integration into Jira from a solution such as Zephyr or qTest. From within the chosen test management repository, IDS sets up testing to succeed with a fully managed process for Test Script Management, test executions & results, smart defects (raise within the test for automated traceability and data logging) and triage process.

> **Building out the basics in quality engineering and test processes increases the ability to deliver automatically generated reporting & dashboard management.**

First, though, it is essential to gather an understanding from stakeholders about what is important to them to report. Business intelligence is a deliverable which must always be clarified and confirmed and never assumed.

It is the quality teams' responsibility to glean this information from the senior team to ensure that testing is being shaped and the data generated from this activity supports the Business Digital Assurance Strategy, development practices and Build & Deployment. All the quality measures will provide a useful insight into more than just the quality of the application. The goal here is to ensure that improvements in software development and deployment processes are also delivered as a result of insights into the levels of quality here too.

Assuring the processes associated with building and deploying applications will force those processes to become more robust. This ensures that when deploying, they are successful and have a catalogue of business intelligence to support it. This is particularly helpful for organizations wanting to deliver a DevOps model into its business.

IDS recognizes the value of testing using production-like data in testing environments and ensuring that a proper testing strategy has been used to create quality test data.

Automation Practices

Automation supporting services set up the ability to, eventually, have a fully automated traceability practice. Moreover, good habits are established, maintained, and actioned without your team even realizing it.

For ultimate efficiency and the ability for the quality engineering process to keep pace with the regular build and deployment processes, one thing is key. Intelligence must be established into testing by delivering an automated testing ecosystem.

This will begin with test data management, the Achilles heel of all testing solutions. Designing and delivering a fully repeatable and automated test data management solution is the first step taken by IDS.

Effective test data management sets up a highly sophisticated quality engineering practice which creates high-quality software that performs competently once it has been deployed. Great test data management ensures tests are created based on live scenarios to capture issues early and prevent problems happening in the future such as bug fixes and rollbacks. Overall, it can help reduce the total cost of change resulting in a fantastically cost-efficient software deployment process.

Furthermore, in a world of GDPR and other data regulations, a test data management solution gives you the chance to garner information. This furthers your data discovery with different test conditions and enhancing your dataset without breaching data regulations.

We recommend either of the following:

1. **Obfuscate the information you need to make it compliant, and then perform a rigorous automated testing process.**
2. **Or, you can create entirely new synthetic data using test data generation and do the same thing - run a testing process and create various testing environments to test your software with data based on reality to quality check how your application reacts.**

Remove human error with an automated solution to support your testing and development and benefit from reliable, continuous test automation without the need to hire consultants and make them do mundane tasks.

Having a continuous quality feedback loop will speed up your real-time deployment process and allow you to have a complete understanding of your software before releasing it.

Decide on what to focus your automated quality engineering solution on. Naturally, the focus is on functional & E2E automated tests. A decision to understand which tests to execute and when is important. The key is to build automated test assets in a way that is modular, this is recognized to hit those early delivered functions and provide early ROI. Using a modular approach, those same tests can be joined as the application builds out and eventually simulates full business workflows end-to-end.

Some of the quickest ways to establish greater ROI and greater coverage from automated testing is to get in earlier and shape your focus to database and API validation testing.

To deliver uplifts in productivity as well as rapid ROI during our deliveries of test automation ensures that we can create our automated test assets. Even when we don't have access to the application or new functionality.

Message-based testing enables you to create templates based on known and predefined elements such as SOAP, REST, JSON, Web Services Description Language (WSDL), message exchange patterns and RESTful services.

Building common and re-usable templates, based on those mentioned above, and communicating effectively between developers and testers alike, automated tests can be written in parallel with application code written by developers.

The IDS Approach

IDS' approach enables users to simulate end-to-end, business process automated tests to use the platform effectively and so automated tests are written in parallel with code. It takes a lot of time and effort upfront to give you a fighting chance of getting it right for your organization. If the combination of people, process and tooling is good then automated testing can deliver rapid returns on investment.

With the correct solution in place, it is possible that your projects will experience a reduction in the time taken in testing cycles as well as the opportunity to have more focus on testing new functionalities.

In fact, given the correct solution, your organization could benefit from enhanced reporting processes, and a significant quality increase around the build process by having a solution which fits into a continuous delivery system.

These factors together give a true measure of quality. By combining the speed of the testing process with enough rigor to demonstrate control of your testing, this gives a good reason to empower your testing team. Automated testing can deliver dramatic results especially if you automate early and often.

Exercise Functionality Efficiently

Historically, automated testing begins at the end of the first release when the UI of the application has reached the point where it is more stable.

At this point, the application will have gone through many cycles of development and manual testing. Especially if when talking about a multi-layer application which has, for example, a Graphical UI, Interface Layer, and a database.

So, with that in mind, why would you wait until the end of the application lifecycle to begin the journey of writing automated tests?

Why Would You Only Focus on Automating The GUI Tests?

Focus on the forgotten layers. Starting at the "forgotten layer" of the automation pyramid, interfaces communicate with each other by sending messages. By considering the two interfaces, the communication pattern would be;

- → **Request**
- → **Response**

The interface requester transmits a message to a destination interface and, in turn, the destination interface returns a response back to the interface requester. There are some exceptions and in some instances, an interface doesn't expect a response.

Each interface is built to expect a standardized message based on a predefined technical definition. Having such a standardized approach presents an opportunity resulting in less ambiguity and more efficient test design.

Message structures rarely change and when they do it is less of an impact as we base everything on a reusable template, so the rate of creating your tests at this layer is rapid, following an initial implementation phase.

Accelerate Quality

Having successfully delivered for some high-profile ERP and PMS organizations using different, carefully selected automated testing solutions from Axe, Selenium, UFT and RFT, the benefits of using accelerator packs were recognized. These are valuable options for newly adopted implementations of SAP, Oracle and Microsoft ERP, PMS and CMS applications.

With a robust approach to test data management and leveraging technologies supporting AI models, customers benefit from an 80% reduction in costs and achieve greater than 85% reductions in risk coverage than a manual and traditional automated testing approach. Thus, resulting in increased coverage and confidence in your delivered solution at go-live.

Using pre-built accelerator packs for automating your tests allows cross-browser support, multiple device testing and ensures that coverage includes a pre-agreed matrix. Using automated test assets, we rapidly test new deployments across multiple handsets, operating systems and other devices. This is from day one of your programs, so straight away your testing is adding value. Then as you configure your implementation, a maintenance approach of your testing assets keeps the pace of development efforts.

Accelerator packs will provide a continuous testing approach straight out of the gate. It is highly recommended as a solution when undergoing a large digital transformation in the ERP, PMS or CMS applications.

Performance Metrics

A well-designed automated testing solution can easily and quickly support Load & Performance by sharing assets such as test scripts and, of course, the test data management process and tooling.

Recent times have brought a real collaboration between automated testing and Load & Performance solutions, with many tooling options being able to convert automation assets into performance assets.

As an example, Tricentis Tosca can convert automated assets into Tricentis Flood assets via the Flood API. The key to creating performance assets is simplicity in the building stage. The skill comes in reading and understanding the results and outcomes of execution. So, using solutions which support cloud-based

Load & Performance testing, with simple load test creation & reuse of existing assets, are key.

Organizations benefit from accessibility & scalability of testing to gain an ability to simulate real browsers. Having a solid engineering structure will reduce costs of scaling your load testing and provide real-time results/visualization of tests for your experts to analyse.

One of the key aspects of performance in the new world of CI/CD and DevOps is having the ability to continuously deliver insight into the performance of your applications.

Continuous Delivery

As software becomes the key to creating competitive advantage across all markets, enterprises no longer enjoy the luxury of selecting either 'speed' or 'quality' when delivering software. Both are critical. Now that agile practices have matured and DevOps initiatives have entered the corporate agenda, continuous integration (CI), continuous testing and continuous delivery (CD) have emerged as key catalysts for enabling quality at speed.

Of the three, continuous testing is by far the most challenging.

With the emergence of continuous deployment and rapid development practices, testing is often considered the bottleneck.

> **"Industry research from Gartner indicates that automated testing levels consistently hover around 20% from a coverage perspective.**
>
> (Source: Gartner)

Even with historic and traditional automated testing solutions in place, testing and quality practices will fall way behind development in a fast-paced program.

Continuous integration is a tool-driven activity for the execution of unit tests etc. Continuous delivery brings in a team element added to CI. Continuous testing brings everything together and involves tools, teams, individuals, and services. The design of what is executed, and when, is critical to the success of having automated business intelligence.

Executed correctly, continuous testing serves as the centerpiece of the agile downstream process. Executing automated and performance tests as part of the software delivery pipeline to provide risk- based feedback as rapidly as possible. Mastering continuous testing is essential for controlling business risk. Presenting the data on quality up to the reporting and dashboards, given the increased complexity and pace of modern application delivery.

Automated Dashboards

Jira configuration is a part of the success of having automated dashboards. The ability to express the levels of quality starts with the gathering of low-level activity which is all fully traceable back to the higher-level artefacts stored in Jira.

We begin with a configuration of Jira based on the criteria explored during our health check, described earlier in this paper. This process sets up the foundational aspects of a solid Jira instance and ensures the work-engine tasks are presenting metrics fit for business intelligence. Integrations into Jira are key here for an insight into quality solutions such as Zephyr, Tricentis qTest or Microsoft Azure. DevOps are a selection of quality focused tools which have brought many successes.

With careful configuration, using the IDS model, the presentation of test outcomes from executions are automatically uploaded against test scripts and bugs have a similar automated approach.

Many organizations are now combining Jira property fields and values with test run data in solutions such as qTest Insights to produce detailed traceability reports. This results in testing teams to better analyze property values related to testing activities.

Mobility is also key, in a world where information is required on-demand we are supporting your stakeholders and the teams to stay ahead.
The ability to monitor progress via dashboards on-demand on your mobile devices is very much a part of our configuration.

Support and Coaching

It is easy to sell a service. But it is very tricky to deliver on that promise which is sold in the boardroom. IDS is proud to provide a different approach to many other similar organizations. IDS are successful in implementations due to the level of involvement. It is not about selling a 300-page manual and having 30-40 consultants sit on your site.

The goal is about enabling and empowering customers and teams to improve and learn how to operate efficiently.

Initial set up, using IDS' A-Players, full support and assistance is provided in which every element of expert service is required. Technicians will work with you to;

→ Conduct a health check

→ Design your solution

→ Implement your solution and coach your team through the process

→ Automate testing (including tool selection)

→ Test data management

→ Test repository set up (including tool selection)

→ Test data management

→ Integration and configure the Jira dashboard.

They are with you for the duration of the project, if required. Ongoing support and guidance allows teams to thrive within a structure for success.

Summary and Benefits

In summary, quality engineering and testing services are used to provide metrics back to the business and stakeholders on the status of the application's quality levels. Using a correct combination of educating teams, correct use and configuration of tooling, good habits and best practice IDS' solutions ensures business intelligence is accurate and readily available.

A full coaching and mentoring service is included to teach you about the solution that is built out to deliver full visibility and accuracy of metrics, providing confidence in decisions. Teams will be able to support multiple, rapid changes to software, with frequent builds and provide a faster turnaround.

Continuous testing with metrics is the cornerstone to evolving your organizations towards sophisticated practices such as continuous deployment and DevOps.

> **By the very nature of delivering in this way, your teams will benefit from improved skills within projects, increased confidence and ultimately reduced spend. Save up to 30% more time because you no longer have to handcraft weekly or monthly reports.**

A high level view of the benefits are seen below;

1. **Good habits happen automatically** – With enhanced testing and traceability, expect people's work standards to rise and the overall testing quality to be improved. An inbuilt process is possible to ensure that as new artefacts are created, then they will automatically be linked back to parent artefacts such as tests, requirements and stories.
2. **Focus on a holistic automated solution** – Instead of looking at just one element of a project at a time, expand your view and look at the entire project from a birds-eye view. This will allow you to see how each facet interacts with one another and ensure you see problems arise.
3. **Enjoy one version of the truth** – Track progress of your high-pressure project and ensure corners are not being cut. Each member of your leadership team can view the status of the project with a bespoke, automated dashboard configuration.
4. **Contract to release and beyond** – Relax knowing every deliverable, from a contact perspective, is being traced to a project artefact.
5. **More varied testing** – With IDS' toolkits, enjoy repeatable, automated testing that can focus on any number of software projects and data sets. Let testing become part of your every day and watch as your application and data quality skyrockets.
6. **Proven and fully integrated** – IDS' methods and products work. By working with a number of massive institutions, bodies and companies, projects were delivered with overall improvements in application and data quality.
7. **Less technical** – IDS are passionate about delivering less technical solutions which open up the benefits of automation to companies and institutions that don't always have internal technical skills to deliver traditional approaches.
8. **Cost-effective** – Focus on the correct amount of testing and reduce your project overheads. Expect a reduction in testing costs by up to 85% and instead repurpose those savings to other areas.
9. **A greater level of visibility** – Expect on-demand reports which can be accessed on several devices. Investors and stakeholders can check in on the status of their project at times that suit them. This visibility will make your projects leaner, increasing ROI because waste is reduced and improves the quality of results.

When data lies at the heart of your organization, you need data certainty. IDS's unique methodology brings together data testing and data quality assurance, to assure 100% of the data, through 100% of the journey, 100% of the time.

To find out more, discuss your data needs with an IDS data certainty specialist here:

James Briers
Co-founder & CTO

T: +44(0)161 240 3603
M: +44(0)776 621 6284
E: james.briers@intelligent-ds.co.uk

Manchester
Office 1:18, Linley House,
Dickinson St, Manchester, M1 4LF

London
WeWork, Moorgate,
1 Fore Street, London, EC2Y 9DT

intelligent-ds.co.uk