Ministry of Defence

EVIDENCE-LED ASSURANCE FOR THE SOFTWARE DEVELOPMENT LIFECYCLE

The Case and Methodology for Continuous Assurance

September 2026

Contents

1. Version Control

Version	Date	Summary of Changes
V0.1	24/07/2026	Initial draft for restricted release and review
V0.2	08/09/2026	Feedback changes clarifying message
V1.0	09/09/2026	Formatting amendments & modified document title to ELA framework
V1.1	18/09/2026	Expanded on Annex E JSP examples + rebrand update

2. Purpose

This paper establishes a methodology to modernise assurance within the software development lifecycle.

It explains how organisations can move from document-based approvals to assurance based on continuous evidence, explicit assertions and confidence-based decisions. The methodology does not remove governance. It makes governance stronger by improving the quality, timeliness and reliability of the information used to make decisions.

It is intended for delivery, assurance, security, service management and governance teams that need to make reliable decisions about changing software systems. A key outcome is shorter lead time to change, achieved by replacing avoidable waiting, rework and document duplication with timely evidence that supports decisions.

The methodology is deliberately not constrained by a particular delivery label or toolchain. Its relevance depends less on how a change is categorised and more on whether that change needs to be evidenced, governed and accepted with appropriate confidence. Where a change affects the assured state of a service, the methodology helps make the basis for that confidence explicit, proportionate and traceable.

The methodology does not replace architecture, cyber security, service management or risk controls. It does not mandate a single toolchain or delivery method. It does not remove human judgement or mean that all evidence must be automated from the start. It establishes a way to make assurance more current, proportionate and useful.

This is Part 1 of a three-part approach. Part 1 establishes the case for evidence-based assurance, defines the core methodology and explains how it strengthens governance. Part 2 defines the assurance standard across the software delivery lifecycle. Part 3 addresses organisational adoption, pilot implementation and transition to wider use.

Readers who need the core argument should read the Executive Summary and Annex D. Readers who need the underlying methodology should read Sections 3 to 10. Readers seeking lifecycle assurance requirements should refer to Part 2. Readers responsible for adoption, pilot delivery and rollout should refer to Part 3.

3. Executive summary

Modern software delivery has changed. Systems are now built, tested, deployed and operated continuously. Governance and assurance now have the opportunity to evolve with them. As delivery becomes continuous, assurance can become more current, evidence-led and responsive. Decisions are strongest when they are supported by current evidence from the system itself.

This paper establishes an evidence-based assurance model that uses five connected concepts to turn delivery data into decision confidence:

- signals;
- metrics;
- evidence;
- assertions; and
- decisions.

Signals and metrics show what is happening. Evidence explains what those observations mean in context. Assertions define what must be true. Decisions are then made according to the confidence that the evidence provides.

The model is deliberately proportionate. Minimum viable evidence defines the smallest credible set of evidence needed to support a specific assertion with sufficient confidence. This prevents assurance from becoming a larger documentary burden while still ensuring that decisions are based on relevant, current and trustworthy information.

The model enables:

- continuous assurance supported by current evidence;
- explicit, testable assertions about system state; and
- confidence-based decisions that reflect evidence quality and remaining uncertainty.

The model retains a simple five-stage lifecycle — define, build, validate, release and operate — and treats those stages as iterative assurance viewpoints. Across those viewpoints, it supports assurance of safety, security, reliability, operational readiness and continued service performance.

Each cycle produces evidence, supports decisions and feeds operational learning into the next change.

The key message is that speed and control reinforce each other when assurance is based on continuous, verifiable evidence. The model reduces lead time to change while improving decision quality and maintaining control.

Similar approaches are already visible in high-assurance public sector environments. The United States Department of Defense has moved towards DevSecOps and continuous authorisation, while the UK Government Digital Service focuses assurance on outcomes, working services and evidence of delivery. These examples show that stronger assurance and faster delivery can coexist when decisions are based on current evidence.

Part 2 translates this methodology into stage-specific assurance expectations, including required observations and authoritative source records, metrics, evidence, assertions and decision criteria across define, build, validate, release and operate. Part 3 defines how the methodology is adopted, piloted and scaled.

4. The problem with current delivery models

Modern software systems change quickly. They are complex, security expectations are high and services often operate continuously. Many governance models still rely on stage gates, static artefacts, periodic reviews and manual approvals.

This creates an opportunity to align assurance with the way modern delivery actually operates. Delivery teams generate change, feedback and operational insight continuously, while governance often checks that work at fixed points using static artefacts.

The operational opportunity is significant. Assurance becomes stronger when decisions are based on evidence that reflects the current system. Timely evidence reduces avoidable delay, prevents repeated explanation and helps teams focus effort on improving the service rather than recreating static descriptions of it. Annex A2 and Annex A3 explain the shift from static, intermittent governance to assurance aligned with continuous delivery.

5. A new operating model

This methodology provides a stronger operating model. It treats software delivery as a connected system, not as a sequence of isolated activities. The central delivery constraint is organisational decision confidence, not simply engineering capability. All subsequent concepts in this methodology exist to improve the quality, timeliness and accountability of that decision confidence. Annex A1 explains how Agile and DevOps support this model: Agile helps teams manage work, while DevOps helps organisations manage delivery as an observable system.

The model draws on the same broad direction seen in other public sector assurance reforms. The United States Department of Defense and the UK Government Digital Service have both moved away from relying mainly on static documentation and towards assurance based on current evidence, defined outcomes and observable system behaviour. Annex C1 and Annex C2 provide summaries of these examples.

The intended improvement is not simply faster delivery or fewer governance meetings. The model should improve measurable outcomes such as lead time to change, change failure rate, service reliability, operational stability and the quality of risk decisions. These outcomes provide the practical test of whether evidence-led assurance is working.

The model introduces five core concepts:

- signals – raw observations generated by systems and processes;
- metrics – structured aggregations of those signals over time;

- evidence – contextualised information used to support or challenge assertions;
- assertions – explicit statements about system state; and
- decisions – actions taken based on the confidence derived from evidence.

Together these form a continuous system:

Signals → Metrics → Evidence → Assertions → Decisions.

The first step is to distinguish between raw observations, aggregated patterns and decision-supporting evidence.

6. Signals, metrics and evidence

Signals, metrics and evidence are related, but they are different. A signal is a raw observation. A metric identifies a pattern or trend across signals. Evidence is created when those observations are interpreted in context to answer a specific question. Annex A4 explains this distinction through a worked example; Part 2 defines the operational requirements for evidence sources, measures, thresholds and ownership.

For example, CPU usage reaching 100% is a signal. It becomes evidence only when considered with duration, normal behaviour, user demand, recent deployment activity, error rates and response times. That evidence can support an assertion about service stability and inform a decision to proceed, proceed with conditions, defer, reject or stop, or escalate. Monitoring or capacity changes are recorded as conditions or required actions.

This distinction matters because many organisations have large amounts of telemetry but still struggle to answer simple questions such as whether a change is safe to release or whether a system is genuinely under control. The opportunity is to convert available telemetry into meaningful evidence that supports accountable decisions.

7. Assertions and the nature of confidence

This methodology makes explicit a layer that is often implicit in delivery governance: assertions. Governance decisions often depend on statements such as ‘this release is safe’, ‘this service is stable’ or ‘this system is compliant’. These are assertions, not just opinions. Each assertion carries an implied burden of proof. Annex A5 explains why assertions need to be explicit in evidence-based assurance.

An evidence-based model makes these assertions clear and testable. Assertions transform governance questions into testable statements. Instead of asking whether a change is acceptable in general terms, the organisation defines what must be true and evaluates whether the available evidence supports that conclusion. This strengthens assurance by defining what the organisation is trying to prove and showing how the evidence supports or challenges it.

The model then supports confidence-based decisions. Confidence does not require complete certainty. Governance decisions routinely operate under uncertainty. The

objective is not to eliminate uncertainty, but to make it visible and determine whether sufficient confidence exists to proceed. Governance becomes more effective when it evaluates confidence, evidence quality and remaining uncertainty, rather than process completion alone. Confidence is not binary. It depends on evidence quality, consistency across observations, metrics and evidence, and remaining uncertainty. Annex A6 explains this decision model in more detail.

The principal decision states are proceed, proceed with conditions, defer, reject or stop, and escalate. These states show whether evidence provides enough confidence, whether conditions or residual risk remain, or whether further review is required. Part 2 defines the precise meaning, evidence conditions, records, authorities and examples for each state.

In a mature model, some low-risk decisions may be automated when agreed evidence thresholds are met. This is an extension of governance, not a replacement for it.

8. Continuous assurance

This enables assurance to operate at the cadence of modern delivery. When evidence was scarce and slow to obtain, organisations had to collect, review and decide periodically. When systems generate evidence continuously, assurance can also become more continuous. Annex A8 describes this as assurance based on ongoing observation and evaluation, rather than isolated review points.

Continuous assurance supports timely decisions while preserving the judgement and accountability needed for higher-risk change. Evidence is perishable. Evidence that accurately described a system at one point in time may no longer describe it after changes in code, configuration, dependencies, operating conditions or threat context. Continuous assurance helps ensure that evidence remains aligned with the current state of the system. It means assurance is supported by evidence that is generated, refreshed and reviewed as the system changes.

observe → evaluate → decide (continuously)

Control is strengthened by embedding it into the delivery system and supporting formal governance where judgement is still required. Some controls become embedded through automated validation, monitoring and continuous verification. Not every decision requires a meeting, but some decisions continue to require formal governance. The model helps both by making the relevant evidence available when the decision is needed.

As evidence maturity increases, low-risk and well-understood changes can move towards automated approval where minimum viable evidence is agreed, controls operate reliably and exception paths remain clear. Annex G introduces this concept.

9. The evidence supply chain

The model depends on a reliable evidence supply chain that generates and presents decision-ready evidence at the point it is needed. Annex A7 describes this as the

evidence supply chain. It is the flow of signals from pipelines, testing tools, deployment platforms, observability systems and security controls into evidence that can support decisions. Build systems produce build and test results. Deployment systems produce deployment outcomes. Observability platforms produce runtime signals. Security tools produce risk indicators. Together, these sources create usable evidence.

The discipline is to collect the right evidence, not more evidence. Minimum viable evidence identifies the evidence needed to support a specific assertion with sufficient confidence. This keeps assurance proportionate by focusing on decision-relevant evidence rather than collecting evidence for its own sake. Annex F provides an example evidence catalogue. Minimum viable evidence is therefore a discipline of focus. It prevents unnecessary evidence collection while ensuring that the evidence gathered remains sufficient to support the assertion being tested.

Documentation may itself provide evidence, but evidence is not synonymous with documentation. The strongest evidence is the most authoritative information available for the decision being made, whether it comes from a controlled document, delivery pipeline, test result, monitoring platform, service record or accountable decision record.

As assertions, evidence requirements, thresholds and decision criteria become embedded in delivery processes, organisations progressively codify knowledge about how their systems operate and how assurance decisions are made. The evidence supply chain therefore serves not only as a mechanism for assurance, but also as a means of preserving, validating and reusing operational knowledge. This reduces reliance on individual expertise and helps ensure that knowledge evolves as the system changes.

Individual pieces of evidence lose value as systems change, but the knowledge captured through assertions, evidence definitions, thresholds and decision records accumulates over time. Evidence is perishable; knowledge is cumulative.

Good evidence is current, relevant, traceable, reproducible, proportionate and attributable to an authoritative source. Weak evidence may be stale, disconnected from the assertion, manually reconstructed without provenance, selectively presented or unsupported by an accountable owner. Part 2 defines how evidence quality, freshness, ownership and retention are assessed in practice.

Modern delivery is both a delivery system and an evidence system. The goal is not just to release software faster. It is to make better decisions, more often, with greater confidence. Annex A9 explains how DevOps enables this evidence-based approach.

The lifecycle provides the structure through which this evidence is organised and reviewed.

10. Lifecycle model

The lifecycle itself remains simple. This paper retains five stages:

- define;
- build;

- validate;
- release; and
- operate.

These stages are assurance viewpoints that help organise evidence across each round of change. The lifecycle is iterative: each change moves through define, build, validate, release and operate, with operational learning feeding the next cycle.

In the define stage, teams set design intent, constraints and assertions. In the build stage, teams create source, infrastructure and repeatable delivery mechanisms. In the validate stage, teams generate evidence about quality, security and readiness. In the release stage, decision-makers judge whether the evidence supports promotion. In the operate stage, teams observe live behaviour and maintain assurance after deployment. Operational learning then informs the next definition of change.

Part 2 defines the assertions, required observations and authoritative source records, metrics, evidence and decision criteria associated with each lifecycle stage.

11. Governance in the new model

Across this lifecycle, governance judges whether the available evidence provides enough confidence to proceed safely and responsibly. The methodology supports that judgement by making evidence clearer, more current and easier to use in decisions.

Architecture, service management, assurance, security and operations remain central. In this model, these functions are supported by better evidence, including live system signals and current information that show how the system is behaving now. Annex A2 explains this governance shift in more detail.

The methodology provides the standardised mechanism for assurance; it does not centralise all authority. Risk thresholds required evidence depth and final acceptance authority remain with the appropriate divisional, functional, contractual or mandated governance bodies. In this sense, the model is federated: it creates a common way to frame evidence and confidence, while allowing accountable authorities to determine what level of confidence is sufficient for their context.

The chain from policy to controls, assertions, evidence and decisions must be explicit. Policy defines the obligation; controls define what must be managed; assertions state what must be true; evidence shows whether those assertions are supported; and decisions record whether sufficient confidence exists to proceed, proceed with conditions, defer, top or escalate.

As the model matures, governance can define a limited automated approval path for low-risk and well-understood changes. Automated approval remains a governed decision path based on explicit evidence, agreed confidence conditions and clear exception handling. It does not remove accountability: evidence outside agreed conditions is routed to human review, with the evidence gap and residual risk made visible.

The purpose is to automate routine evidence checks where confidence is high, allowing governance attention to focus on exceptions, contradictory evidence, repeated control failures, emerging risk and continued reliability of the evidence model. Part 2 defines eligibility, thresholds, evidence freshness, exception conditions, review routes and audit requirements.

12. Risks and safeguards

The methodology depends on safeguards that preserve evidence quality, human judgement and accountable decision-making. Part 2 defines the detailed evidence standards, controls and exception mechanisms used to apply them.

The principal risks are false confidence from weak or incomplete evidence, teams optimising metrics rather than outcomes, inconsistent evidence across teams or suppliers, automated checks being treated as a substitute for judgement, and technical evidence being considered without sufficient operational context.

These risks are managed by making assertions explicit, defining minimum viable evidence, agreeing evidence-quality standards, keeping uncertainty and residual risk visible, and retaining accountable ownership for every decision. Evidence informs judgement; it does not replace it.

Where confidence is insufficient, the decision records the evidence gap, residual risk, accountable owner and required outcome: proceed with conditions, defer, reject or stop, or escalate for further judgement. Automated approval is used only when the evidence supply chain is reliable, agreed confidence conditions are met and exceptions return to accountable human review.

13. Transition to implementation

Implementation begins through controlled adoption alongside existing governance. The initial objective is to test whether assertions can be made explicit, evidence can be obtained reliably and confidence-based decisions improve clarity, timeliness and control without weakening accountability.

Successful adoption will require investment in the assurance-engineering interface. This includes the tooling needed to surface evidence, the skills needed to interpret it, the process integration needed to connect delivery and governance, and the operating agreements that define ownership, thresholds, exceptions and decision routes. Without this interface, evidence-led assurance risks becoming an additional reporting layer rather than a more effiective way of governing change.

Part 2 defines the lifecycle assurance standard, including mandatory assertions, evidence requirements, decision states and exception mechanisms. Part 3 defines how organisations assess readiness, select and govern a pilot, measure outcomes and transition to wider adoption.

14. Conclusion

Modern software delivery requires assurance that is as current, observable and adaptive as the systems it governs.

Static artefacts remain useful for traceability, record keeping and context, but they are strongest when complemented by continuous evidence. Continuous evidence gives decision-makers a current, verifiable and proportionate basis for judgement. Assertions make the burden of proof explicit. Confidence-based decisions provide a more realistic model than binary pass-or-fail process completion.

The central point is simple:

Continuous evidence strengthens control by giving decision-makers relevant empirical data when it is needed.

This paper establishes the methodology and case for change. Part 2 defines the assurance standard across the software delivery lifecycle, including what good looks like, the assertions to be tested, the required observations and authoritative source records, derived metrics, evidence and decision criteria used to determine whether sufficient confidence exists to proceed. Part 3 defines how organisations adopt, pilot and scale the methodology.

Annex A
Purpose

This annex provides short explanations of the key concepts introduced in the main paper.

Summary

- **A1–A3:** Agile and DevOps create observable delivery, while governance evolves to use evidence at the pace of continuous change.
- **A4–A6:** Signals and metrics become evidence when interpreted in context, tested against explicit assertions and used to support confidence-based decisions.
- **A7–A9:** A reliable evidence supply chain enables continuous assurance and establishes DevOps as both a delivery system and an evidence system.

Key point

Observable delivery, contextualised evidence and explicit assertions combine to enable continuous, confidence-based assurance throughout the lifecycle.

Annex A1: Agile and DevOps in a modern delivery model

Purpose

This annex explains the roles of Agile and DevOps in modern software delivery.

Summary

Agile improved how teams build software. It introduced iteration, collaboration and early delivery of working products. This allowed teams to respond quickly to change and deliver value sooner.

As software became more important, organisations faced a wider challenge. They had to coordinate work across many teams, manage dependencies, govern change and operate services at scale. Agile was not designed to manage these system-level concerns on its own.

DevOps addresses this gap. It focuses on the whole delivery system rather than individual teams. It helps organisations understand how work flows from idea to production. It also helps them understand how systems behave once they are live, and how to improve them over time.

Key point

Agile helps teams manage work. DevOps helps organisations manage delivery as a system. A modern lifecycle needs both.

Annex A2: Governance in a modern delivery environment

Purpose

This annex explains how governance needs to change to support modern delivery.

Summary

Most organisations already have strong governance. They use reviews, approvals and controls to manage risk and maintain stability. These controls are necessary.

The opportunity is to make governance more current by complementing documents, reports and status updates with live evidence from the system itself.

This reduces delay and duplication by allowing teams to focus more effort on improving the system and less effort on recreating evidence manually.

Modern delivery systems generate a stream of signals. These include delivery metrics, monitoring data and operational indicators. These signals show how the system is performing in real time.

Key point

Governance is strengthened when live system signals inform decisions, making control more accurate, timely and relevant.

Annex A3: Continuous delivery and intermittent governance

Purpose

This annex explains the gap between modern delivery and traditional governance.

Summary

Traditional lifecycle models were built for a different environment. Delivery was sequential. Releases were infrequent. Evidence was slow to collect. In that context, stage gates and formal approvals were effective ways to control risk.

Modern delivery is continuous. Changes are frequent and small. Feedback is rapid. Testing, security and operations happen together.

Continuous delivery is also iterative. Each change should move through define, build, validate, release and operate. Learning from operation should then inform the next change. The model is therefore a repeatable cycle, not a one-off sequence.

Governance can now evolve to match the cadence and evidence flow of modern delivery. Rather than depending only on fixed phases and point-in-time evidence requests, it can use useful information produced by the system throughout the lifecycle.

The opportunity is to align governance with the rhythm of delivery, so decisions use current system behaviour rather than static snapshots.

Key point

Governance becomes more effective when it moves from periodic checks to continuous assurance based on real system behaviour.

Annex A4: Signals, metrics and evidence

Purpose

This annex defines the difference between signals, metrics and evidence.

Summary

Modern systems produce large amounts of data about builds, deployments, performance and failures. The opportunity is to turn that data into evidence that answers practical questions about safety, reliability and readiness.

This requires a clear distinction between signals, metrics and evidence.

A signal is a raw observation. Examples include a test result, a failed deployment or a change in response time. Signals are useful starting points that become more powerful when interpreted with context.

Metrics are patterns drawn from signals over time. They show trends such as error rates or deployment frequency. Metrics provide useful context and become decision-supporting evidence when interpreted against the question being answered.

Evidence is different. Evidence is formed when signals and metrics are combined, interpreted and used to answer a specific question.

For example, CPU usage reaching 100% is a signal, but it does not show whether the service is at risk. A sustained increase from its normal range becomes a metric. When interpreted alongside user demand, recent deployment activity, autoscaling behaviour, error rates and response times, it becomes evidence for an assertion about service capacity and stability. Part 2 extends this example by defining the authoritative source, required metric, locally agreed threshold, evidence owner and decision conditions.

Example flow

- signal: CPU usage reaches 100%;
- metric: CPU usage remains materially above its normal range;
- evidence: load, response time, error rate, demand and recent change data show whether capacity is adequate;
- assertion: the service has sufficient capacity and remains stable; and
- decision: proceed, proceed with conditions, defer, reject or stop, or escalate; monitoring or capacity changes are recorded as conditions or required actions.

Key point

Decisions are strongest when they are based on interpreted evidence, not isolated signals or headline metrics.

Annex A5: Assertions and assurance

Purpose

This annex explains the role of assertions in governance.

Summary

Evidence supports decisions most effectively when the organisation is clear about what it is trying to prove.

Statements such as “this release is safe” or “this system is stable” are common. These are not opinions. They are assertions.

Each assertion requires proof. Evidence-based assurance makes that proof more current, testable and transparent by linking decisions to evidence generated by the delivery system.

Key point

Clear assertions link evidence to decisions. They make governance explicit, measurable and reliable.

Annex A6: Confidence-based decisions

Purpose

This annex explains how decisions are made in an evidence-based model.

Summary

Traditional governance relies on approvals and checkpoints because real-time insight has often been limited. Evidence-based governance strengthens this model by adding current evidence to the decision.

In an evidence-based model, the focus changes. The key question becomes whether there is enough confidence to proceed.

Confidence depends on evidence quality, consistency across observations, metrics and evidence, and the level of remaining uncertainty.

Decisions become more nuanced. Instead of approve or reject, decisions reflect levels of confidence.

The principal states are proceed, proceed with conditions, defer, reject or stop, and escalate. They express whether confidence is sufficient, conditional, incomplete, unacceptable or beyond delegated authority. Part 2 defines the exact conditions and examples for each state.

As evidence maturity improves, some low-risk decisions can become suitable for automated approval where the required evidence, thresholds and exception paths have been agreed in advance.

Key point

Governance is strengthened when it evaluates confidence, not just process completion. This allows decisions to be made continuously as evidence becomes available.

Annex A7: The evidence supply chain

Purpose

This annex explains how evidence is created and managed.

Summary

Evidence depends on signals produced throughout the delivery lifecycle. These signals come from pipelines, testing tools, monitoring systems and security controls.

When these sources are connected, they form an evidence supply chain. This ensures that information is generated, captured and made available in a consistent way.

With this supply chain, evidence becomes more consistent, current and reusable. Decisions can rely more on observable facts and less on manual reconstruction.

The supply chain also supports minimum viable evidence. Teams identify the evidence needed to support specific assertions and make that evidence reliable, current and traceable.

Key point

A reliable evidence supply chain gives decision-makers accurate, current and reusable information at the point of decision.

Annex A8: Continuous assurance

Purpose

This annex explains the shift from periodic assurance to continuous assurance.

Summary

Scheduled reviews and approval forums remain useful governance mechanisms, and continuous evidence enables them to operate with more current insight.

Modern systems produce evidence continuously. Continuous assurance uses that evidence to support timely decisions while preserving the judgement and accountability needed for higher-risk change.

It changes how organisations operate by basing assurance on ongoing observation and evaluation rather than isolated review points.

Control remains present and becomes more embedded. Some controls operate through automated checks, monitoring and verification.

Key point

Assurance is strongest when it is continuous and integrated into delivery, rather than separate from delivery and performed only at fixed intervals.

Annex A9: DevOps as an evidence system

Purpose

This annex explains the role of DevOps in enabling evidence-based delivery.

Summary

DevOps is often described in terms of tools such as pipelines and automation. These tools are valuable because they help create the observable delivery system needed for evidence-based decisions.

The main value of DevOps is that it connects delivery, testing, operation and monitoring into a single, observable system that can generate and use evidence.

With this capability, signals are continuous, evidence is timely and decisions can be based on how the system is actually behaving.

Key point

Modern software delivery is both a delivery system and an assurance system because it continuously generates evidence that supports accountable decisions. DevOps enables this capability.

Annex B: Preview of what good looks like

Purpose

This annex provides a high-level preview of the practical characteristics of good evidence-based assurance. Part 2 defines stage-specific expectations and evidence patterns in detail.

Summary

Good evidence-based assurance starts with clear assertions. Assertions define what must be true about the system. Examples include whether a release is safe, whether a service is reliable and whether required controls are operating as intended.

Teams define what good looks like for each assertion. This includes the expected outcome, the minimum viable evidence needed to support it, the acceptable level of uncertainty and the decision states that may apply.

Evidence should come directly from the delivery system and running service where possible. At a conceptual level, it may include test, deployment, security, performance and operational information. Part 2 defines the mandatory evidence sets, sources, freshness rules and owners for each lifecycle stage.

Evidence quality matters. Good evidence is current, relevant, traceable, reproducible, proportionate to the decision and attributable to an authoritative source. Weak evidence can create false confidence, even when it appears objective.

What good looks like extends beyond release readiness. It includes evidence that the service can be operated, monitored, recovered and improved safely. Resilience and recovery are therefore part of the assurance model throughout the lifecycle, not separate activities performed after delivery.

Part 2 defines stage-specific assertions, evidence expectations and confidence conditions, including detailed resilience, recovery and operational-readiness requirements.

Good governance makes risk visible and supports informed judgement. It helps teams move faster by using empirical evidence to reduce avoidable waiting, rework and repeated explanation.

Key point

Good looks like clear assertions, defined minimum viable evidence, current system data, visible uncertainty and accountable decisions.

Annex C: Case studies

Annex C1: United States Department of Defense

The United States Department of Defense provides a useful example of how a high-assurance organisation changed its delivery and assurance model to reduce delay and strengthen control.

Historically, the DoD relied on a traditional approval process based on formal authority to operate decisions, extensive documentation and manual assessment. Over time, this created friction because systems could evolve after approval, security checks were often late, and even small changes could trigger disproportionate re-approval overhead.

The shift to a DevSecOps-based model and continuous authorisation changed how control was applied. Instead of relying mainly on documentation created at fixed points in time, assurance was increasingly supported by evidence generated automatically during build, validation, deployment and operation. This included pipeline evidence, security results, infrastructure checks and operational monitoring.

This example matters because it shows that a high-assurance environment can adopt continuous, evidence-based assurance without removing control. It demonstrates that strong governance and faster delivery can coexist where systems continuously generate reliable evidence of their state and the controls operating upon them.

The DoD reference design describes DevSecOps as a lifecycle that automates, monitors and applies security across planning, development, build, test, release, delivery, deployment, operation and monitoring. It links faster delivery with stronger control by shifting testing and security earlier, using automated checks and maintaining continuous visibility of risk.

The continuous authorisation model develops this further. It expects a software factory and associated teams to support continuous monitoring of controls, active cyber defence and ongoing risk reporting. This does not remove the authorising official or the need for governance. It changes the basis of authorisation from a static approval event towards continuous risk awareness.

Platform-based approaches, such as Platform One, are relevant because they show how common tooling, hardened components and repeatable delivery patterns can reduce accreditation and delivery friction. The important lesson is not that a specific platform should be copied. It is that common evidence patterns can make assurance faster, more consistent and easier to govern.

Relevant public sources include the DoD Enterprise DevSecOps Reference Design, the DoD DevSecOps Continuous Authorization Implementation Guide, Platform One material and NIST SP 800-218 Secure Software Development Framework. These sources show a consistent direction: secure software delivery should embed automated testing, security, monitoring and risk evidence throughout the lifecycle, rather than treating security approval as a late or separate event.

Annex C2: UK Government Digital Service

The UK Government Digital Service provides a second relevant example from the public sector.

Historically, public sector delivery often depended on extensive upfront design documentation, centralised approvals and sequential delivery phases. These approaches aimed to provide structure, but they often resulted in slow delivery and artefacts that no longer reflected the live system by the time decisions were taken.

The Government Service Standard introduced a different assurance model. Instead of treating documentation volume as the main measure of quality, it defined the outcomes that services needed to demonstrate. Teams are assessed on evidence such as working software, operational capability, user feedback, accessibility, security and service performance. This reduced the dependency on static documentation as the main basis for assurance and supported more iterative delivery.

This example is relevant because it shows outcome-based assurance in a regulated UK public sector environment. It demonstrates that iterative delivery, lighter artefacts and strong assurance can work together when the focus is on outcomes and evidence of delivery rather than document volume alone.

The service assessment approach is useful because it does not treat documentation volume as the primary measure of quality. Teams are expected to show how the service works, how it meets user needs, how it is operated, how risks are managed and how the team will continue to improve it. This is close to the principle behind evidence-based assurance: judgement is based on current evidence of outcomes, supported by artefacts where useful, rather than on artefacts alone.

Relevant public sources include the GOV.UK Service Standard, GOV.UK service assessment guidance and published service standard reports. These sources show that assurance can focus on evidence of user need, working services, operational readiness, security, accessibility, performance and continuous improvement.

- the GOV.UK Service Standard defines the outcomes teams must meet when building and running services;
- service assessments test whether a service meets those outcomes at key lifecycle stages; and
- published assessment reports show that decisions can be expressed as green, amber or red, with actions required where evidence is incomplete or risks remain.

Annex D: Short briefing summary

Purpose

This annex provides a short briefing version of the paper for readers who need the main message, recommendation and next step.

Briefing summary

Modern software delivery is continuous, iterative and highly observable. Governance and assurance can become more effective when periodic review and document-led assurance are complemented by current evidence from the delivery system.

This paper establishes an evidence-based assurance model built around signals, metrics, evidence, assertions and decisions. The model turns delivery data into decision confidence, reduces lead time to change, improves decision quality and maintains control.

The next step is controlled adoption through a suitable pilot. Part 3 defines how the pilot is selected, scoped, governed and measured. The pilot applies the assurance standard defined in Part 2 while existing mandatory governance remains in force.

Part 1 establishes the methodology and case for change. Part 2 defines its practical assurance requirements across the lifecycle. Part 3 provides the adoption, pilot and rollout approach.

Key message

Speed and control reinforce each other when assurance is based on continuous, verifiable evidence. The methodology gives decision-makers better evidence earlier, so decisions can be faster, clearer and more reliable.

Annex E: Policy alignment

Purpose

This annex explains the principle of aligning the methodology with relevant policy. Part 2 defines the required structure for mapping controls, assertions, minimum viable evidence and residual-risk ownership. Part 3 defines how that mapping is introduced, governed and maintained during adoption.

Summary

The methodology is aligned with the direction of UK Government Secure by Design policy and the MOD Secure by Design approach. Both expect security and cyber-risk management to be integrated throughout the service or capability lifecycle, with clear accountability, early identification of threats and weaknesses, continuous risk management, defined controls, supply-chain engagement, and assurance, verification and testing.

This alignment does not by itself demonstrate compliance. Compliance is established when the applicable policy and contractual requirements are mapped to explicit assertions, authoritative evidence, accountable owners and retained decision records. Part 2 defines the practical structure for completing that mapping.

The methodology should therefore be treated as mechanism for demonstrating compliance, not as an alternative to mandated compliance or accreditation routes. Where existing policy, contractual or accreditation processes require formal approval, the model strengthens those processes by providing clearer evidence, explicit assertions and retained decision records.

MOD-specific compliance. For MOD capabilities, the applicable requirements may include Secure by Design policy, relevant provisions of various JSP's such as 453, 440, 441, 901, 935, Industry Security Notices, contractual Security Aspects Letters, Defence Conditions, Defence Standards and other mandated assurance requirements. The exact obligations must be confirmed for the capability and contract concerned.

The purpose of the model is to strengthen the evidence available to those controls. It helps governance bodies make decisions using current empirical data, rather than relying mainly on static documents and retrospective explanation.

Policy alignment identifies which controls require evidence, which assertions support those controls, what minimum viable evidence is needed and who is accountable for accepting residual risk. Where policy requires formal approval, the model supports that approval with better evidence.

Key point

Evidence-based governance strengthens policy compliance by making relevant evidence easier to obtain, review and trace.

Annex F: Illustrative evidence catalogue

Purpose

This annex illustrates how evidence can be organised across the lifecycle. It demonstrates the catalogue concept rather than defining mandatory evidence requirements. Part 2 provides the stage-specific assertions, required observations and authoritative source records, metrics, evidence expectations, confidence conditions and ownership needed to apply the assurance standard.

Summary

An evidence catalogue connects each assertion to the observations, authoritative source records, derived metrics and minimum viable evidence needed to support a decision without mandating a single toolchain.

Illustrative evidence types include:

- define: evidence of intent, constraints, risk and required outcomes;
- build: evidence of traceability, repeatability, dependency control and engineering quality;
- validate: evidence that quality, security, performance and readiness assertions have been tested;
- release: evidence supporting promotion, recovery, support readiness and accountable risk acceptance; and
- operate: evidence of live service behaviour, resilience, incidents and continual improvement.

These examples demonstrate the pattern rather than prescribe a fixed checklist. Each service selects evidence that is proportionate to its risk, criticality, architecture and operating context. Part 2 defines how evidence is linked to assertions, confidence conditions, decisions and accountable owners.

Key point

An evidence catalogue connects assertions to proportionate evidence, helping decision-makers understand both the basis for confidence and any remaining evidence gaps.

Annex G: Automated approvals

Purpose

This annex introduces automated approval as a future maturity concept within evidence-based governance. Part 2 defines the detailed eligibility criteria, thresholds, exception logic, human review paths and audit requirements.

Summary

Automated approval is introduced only when the organisation has confidence that the evidence supply chain is reliable, minimum viable evidence is consistently met and relevant controls operate as intended.

It applies only to defined, low-risk and well-understood change types. Detailed eligibility criteria, thresholds, freshness rules, exception paths, human review and audit requirements are established in Part 2.

Key point

Automated approval is governance applied through agreed evidence, thresholds and exception handling, so teams can focus on risk, weak signals, repeated exceptions and continuous improvement.