

WIPO ICT Leadership Dialogue

WILD/2

Geneva, April 14–16, 2026

TDD Competence for Sustainable AI-First Public Services

Building trust, speed and quality
in digital government with
Test-Driven Development

Aleksej PERIĆ
Head, IT Division
IPO, Republic of Serbia

Overview and Purpose

Why TDD for AI-first IPO

AI accelerates delivery but adds evolving risks and uncertainty. Test-Driven Development (TDD) sets clear expectations before deployment.

Predictability and Debt Reduction

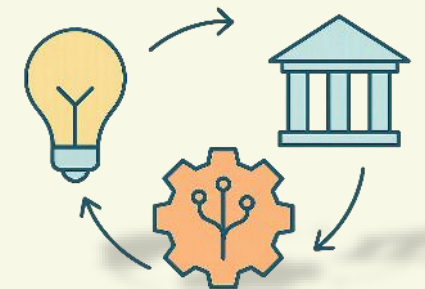
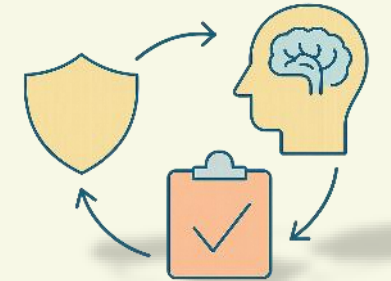
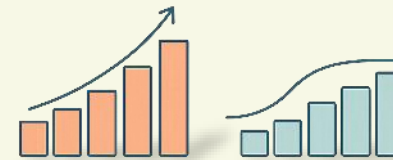
Defining tests up front drives stable behavior and curbs unmanageable technical debt as systems evolve.

Trust, Compliance, Continuity

Reliable AI workflows sustain public trust, legal certainty, and the Intellectual Property Office (IPO) operations.

Purpose of This Presentation

Show how embedding TDD ensures AI-first transformation remains sustainable, compliant, and strategically aligned.



Beyond TDD: BDD, DDD

Test-Driven Development - How to build *thing right*?

TDD (Test-Driven Development) focuses on writing unit tests before code.

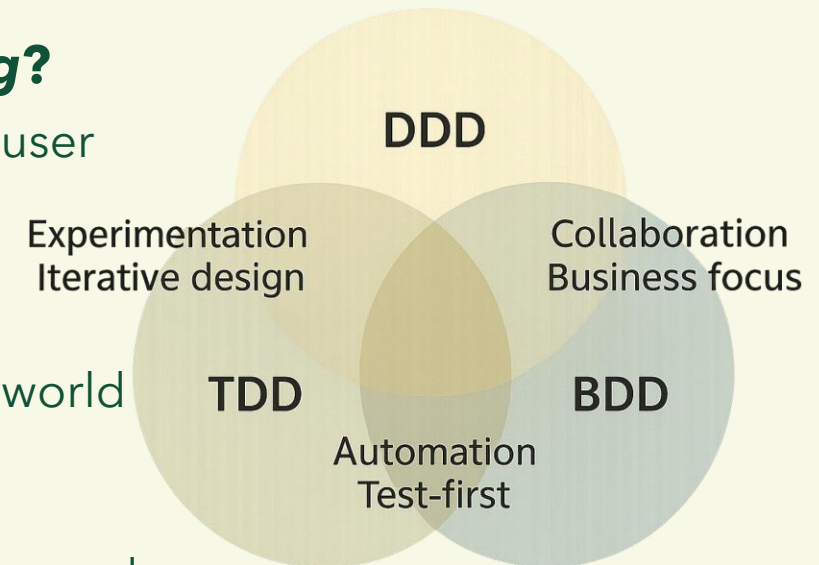
Behavior-Driven Development - How to build the *right thing*?

BDD (Behavior-Driven Development) defines system behavior through user scenarios in natural language.

Domain-Driven Design - The big picture

DDD (Domain-Driven Design) models complex software based on real-world business domains.

These are complementary software development approaches, optimally used together. While mentioning TDD, the other descriptive methodologies are considered as well, all aligned with the legislation, standards and quality management system



Acceleration and Structural Fragility

Acceleration: real, material upside

AI-first methods can unlock rapid innovation, elevate examiner productivity, and improve citizen-facing services through classifiers, summarizers, and search augmenters.

Fragility: volatility in core processes

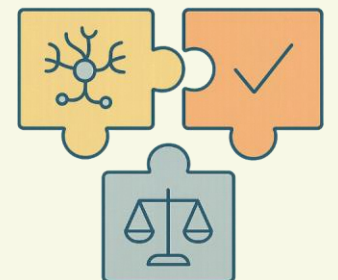
Unlike deterministic code, models evolve with data and tuning, introducing drift, inconsistent outputs, and integration brittleness that threaten accuracy, consistency, and explainability.

Why TDD matters for the IPO

Test-Driven Development (TDD) defines expected behaviors, validates them continuously, and curbs uncontrolled variability before it propagates into downstream workflows and service quality.

Executive takeaway

Acknowledge both upside and risk. Pair AI acceleration with rigorous engineering controls to ensure dependable, explainable, high-quality services.



Unmanaged Risk & Rapid Tech Debt

AI-first adoption without engineering discipline harms stability

Prototype enthusiasm → production instability

AI prototypes (e.g., automated patent classification, examiner support) may skip rigorous testing. When pushed to production, gaps in coverage and reliability surface under real workloads.

No TDD → hotfixes and brittle systems

Without tests, teams rely on prompt-level patches, partial data refreshes, and unverified integrations – creating unpredictable behavior and compounding uncertainty with every change.

IPO impact: inconsistency erodes trust

Inconsistent classification leads to misrouted applications, examiner rework, process inefficiencies, and diminished confidence from applicants and stakeholders.

Short-term speed vs. long-term resilience

Naive AI-first delivery accelerates timelines but undermines stability, inflates costs and risks reputational integrity as tech debt compounds exponentially.



Structured, Predictable, and Validated AI Evolution

TDD shifts AI-first development from reactive and unstable to disciplined and predictable by defining expected outcomes before implementation.

Behavioral boundaries over determinism

Define constraints, data integrity checks, prompt validations, and golden datasets to ensure consistent, auditable performance.

IPO validation before deployment

Predefine classification accuracy thresholds, summarization patterns, and repeatable decision-support logic.

Lower risk, higher confidence

The loop—write tests, implement features, validate behavior, automate pipelines — yields stable, maintainable AI modules that evolve safely.



How TDD approach is adapted for AI

Instead of classic TDD, AI-first teams often combine:

- Evaluation-driven development (metrics first, not tests)
- Data validation tests (schema, distribution, drift)
- Acceptance tests using ranges, not exact matches
- Offline evaluation pipelines
- Canary releases (to a small subset of users before a full release) and monitoring
- Human review loops

Classic TDD has more value *around* the AI (than *inside* it), mostly for:

- Data pipelines
- Feature extraction
- Model orchestration
- Fallback logic
- Safety checks



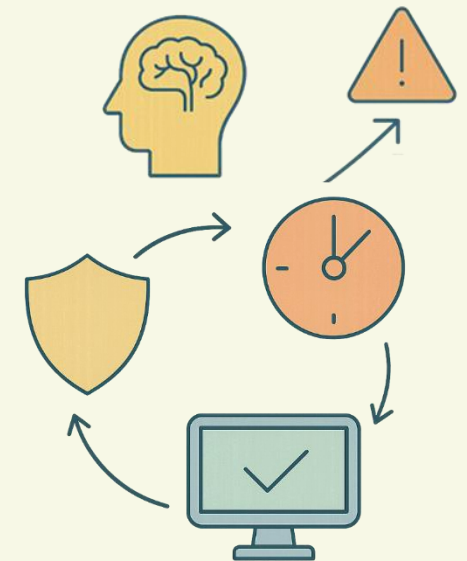
TDD versus AI-First

Classic TDD assumes correctness is defined, stable, and deterministic.
AI-first systems *violate all three assumptions*.

Used blindly, TDD can slow AI development, hide real risks, and create false confidence.

Used selectively—around infrastructure and guardrails—it remains valuable.

Hybrid AI Architecture ensures the core business logic remains predictable, compliant, and auditable while leveraging AI for specific high-value tasks. By treating AI components as "satellites"—isolated microservices accessible via APIs—you maintain modularity, making them testable and replaceable.



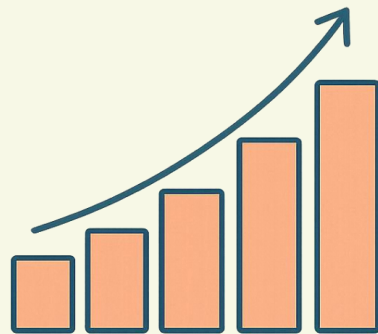
Long-Term Cost and Stability Implications

Naive AI-First Adoption

Technical debt accelerates as features land on an unstable baseline, driving firefighting, rework, and integration failures.

5-year trajectory (relative debt):
10 → 25 → 50 → 90 → 150.

Outcome: rising maintenance cost, inconsistent behavior, elevated risk.

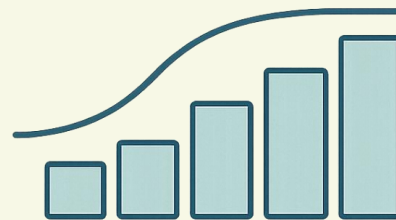


TDD-Driven Adoption

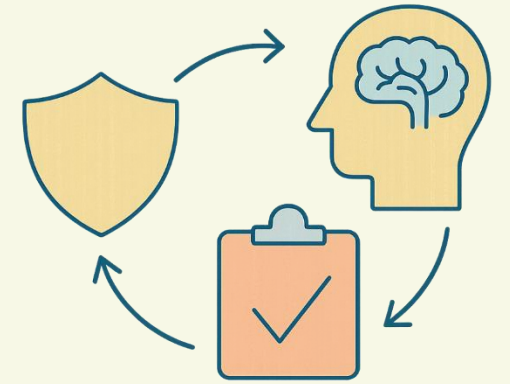
Behavior is validated upfront, keeping debt growth slow, controlled, and predictable.

5-year trajectory (relative debt):
11 → 14 → 17 → 19 → 20.

Outcome: lower maintenance cost, reduced risk, stronger service quality—key for IPO readiness.



Strategic Investments in Skills, Processes, and Governance



Building TDD competence is an organizational capability – not just a technical initiative.

- Prioritize upskilling in test thinking, AI-specific quality assurance, and behavior-driven validation across teams and disciplines.
- Automated testing frameworks and data-quality pipelines enable continuous reliability and rapid, safe iteration.
- Enforce accountable architecture decisions, risk protocols, and clear policies for AI output validation and oversight.
- Global IP office examples show stronger governance and disciplined engineering yield higher AI reliability, smoother audits, and greater stakeholder trust.

Strategic Imperatives for IPO IT Leaders

Modernization without systematic testing elevates risk, cost, and uncertainty. Embedding TDD as a core capability delivers stability, trust, and long-term system resilience.

Sustainable AI adoption is not just deploying models—it is establishing the structures, feedback loops, and competencies that make those models reliable in production.

Given IPO standards for auditability, accuracy, and fairness, a TDD-driven approach strengthens evidence, reduces bias risk, and supports verifiable changes over time.

Call to action: invest in skills, enforce disciplined processes, and fortify governance to build future-proof public services that responsibly integrate AI advancements.

