IOM Starter Kit

Infrastructure Operating Model — Practitioner's guide to maturity, tool mapping, and adoption planning

Published by theIOM.org · Aligned to IOM Standard v1.0 (proposed open standard — draft)

For practitioners, architects, and infrastructure leaders evaluating IOM readiness in their organization.

How to use this kit

This starter kit is a working document. It is designed to be completed by infrastructure and platform leaders — individually or as a facilitated team exercise — and to produce a concrete, organization-specific picture of IOM readiness and a structured path forward.

What this kit contains

- **Part 1 — Maturity Assessment** Score your organization across the six IOM capability dimensions. Identify where you are today.
- **Part 2 — Tool Mapping Worksheet** Map your existing tools against IOM requirements and surface the structural gaps.
- **Part 3 — Gap Analysis** Translate your maturity scores and tool gaps into a prioritized gap inventory.
- **Part 4 — Adoption Roadmap** Plan your IOM adoption in four phases with milestones, owners, and success criteria.
- **Part 5 — IOM Conformance Checklist** Evaluate any solution claiming IOM capability against the standard's mandatory requirements.

Recommended facilitation approach

This kit works best as a structured half-day workshop with 4–6 participants spanning infrastructure engineering, security, platform, and architecture. The maturity assessment (Part 1) typically takes 60–90 minutes and generates the most valuable organizational alignment. Parts 2–4 can follow in a second session.

- Assign a facilitator to read each capability dimension aloud and lead scoring discussion
- Record dissenting views — disagreement on scores reveals organizational ambiguity about ownership and intent
- Do not score aspirationally — score where the capability reliably exists today, not where it exists in isolated pockets
- Complete Part 3 (Gap Analysis) immediately after Part 1 while scores are fresh

> **Scoring discipline** The most common mistake in maturity assessments is aspirational scoring. Score the capability as it reliably exists across your environment — not as it exists in your best team, your newest platform, or your most recent initiative. Gaps scored accurately are gaps you can close. Gaps scored generously are risks you carry invisibly.

PART 1

IOM Maturity Assessment

Score your organization on each of the six IOM capability dimensions using the scale below. Each dimension is assessed across four sub-criteria on a 0–3 scale (None → Partial → Defined → Governed), for a maximum of 12 points per dimension and 72 overall.

Score	0 — None	1 — Partial	2 — Defined	3 — Governed	
Meaning	Capability does not exist or is entirely manual/tribal	Exists in isolated pockets; inconsistent across teams or environments	Documented, repeatable, and applied consistently across most environments	Continuously operating, machine-enforced, and improvement loops are in place	*Your score*

Dimension 1 — Infrastructure State Modeling

Does your organization maintain an authoritative, continuously updated model of what infrastructure exists, how it is connected, and what its current configuration is?

Capability	0	1	2	3	Max	Score
Asset discovery and inventory All infrastructure assets are discovered and inventoried automatically	0 — None	1 — Partial	2 — Defined	3 — Governed	/3	*Score:*
Relationship and dependency modeling Service dependencies, network paths, and trust relationships are explicitly modeled	0 — None	1 — Partial	2 — Defined	3 — Governed	/3	*Score:*
Configuration accuracy Actual running configuration is continuously captured and reconcilable	0 — None	1 — Partial	2 — Defined	3 — Governed	/3	*Score:*
Cross-domain coverage Modeling covers cloud, network, identity, and on-premises environments as a unified view	0 — None	1 — Partial	2 — Defined	3 — Governed	/3	*Score:*
SUBTOTAL — State Modeling					**/12**	*Total:*

Dimension 2 — Intent and Blueprint Governance

Is the intended design and behavior of your infrastructure explicitly defined, versioned, and machine-readable — rather than living in human memory, documentation, or configuration code?

Capability	0	1	2	3	Max	Score
Explicit architectural intent Intended infrastructure behavior and design constraints are documented in a machine-readable, versioned form	0 — None	1 — Partial	2 — Defined	3 — Governed	/3	*Score:*
Blueprint reusability Approved architecture patterns are encoded as reusable blueprints — not IaC templates — that govern multiple environments	0 — None	1 — Partial	2 — Defined	3 — Governed	/3	*Score:*

Capability	0	1	2	3	Max	Score
Constraint enforcement Architectural constraints (segmentation, data boundaries, access limits) are enforced automatically, not just documented	0 — None	1 — Partial	2 — Defined	3 — Governed	/3	*Score:*
Blueprint versioning and attribution Blueprints are versioned, attributed to owners, and changes are controlled and auditable	0 — None	1 — Partial	2 — Defined	3 — Governed	/3	*Score:*
SUBTOTAL — Intent and Blueprints					**/12**	*Total:*

Dimension 3 — Continuous Reconciliation

Does your organization continuously compare what infrastructure is intended to be, what it was built to be, and what it is actually running as — and act on meaningful deviations before they become incidents?

Capability	0	1	2	3	Max	Score
Built vs. running state comparison Designed (built) state and actual running state are compared continuously — not periodically	0 — None	1 — Partial	2 — Defined	3 — Governed	/3	*Score:*
Drift classification Drift is classified by severity and domain impact, not just flagged as a binary deviation	0 — None	1 — Partial	2 — Defined	3 — Governed	/3	*Score:*
Pre-execution validation Proposed changes are validated against intended state before execution — not detected as drift after the fact	0 — None	1 — Partial	2 — Defined	3 — Governed	/3	*Score:*
Reconciliation frequency Reconciliation operates in real time or near-real time — not on a daily/weekly schedule	0 — None	1 — Partial	2 — Defined	3 — Governed	/3	*Score:*
SUBTOTAL — Continuous Reconciliation					**/12**	*Total:*

Dimension 4 — Ownership and Accountability

Is ownership of infrastructure elements, blast radius, and decision authority explicitly encoded and machine-readable — not reliant on tribal knowledge or manual lookup?

Capability	0	1	2	3	Max	Score
Explicit infrastructure ownership Every infrastructure element has an explicit, machine-readable owner — not just a team name in a wiki	0 — None	1 — Partial	2 — Defined	3 — Governed	/3	*Score:*
Blast-radius awareness The downstream impact of any change is knowable before execution — not reconstructed during incidents	0 — None	1 — Partial	2 — Defined	3 — Governed	/3	*Score:*

Capability	0	1	2	3	Max	Score
Decision authority mapping Who may authorize changes, exceptions, and escalations is explicitly defined and enforced	0 — None	1 — Partial	2 — Defined	3 — Governed	/3	*Score:*
Accountability during incidents When something breaks, ownership is immediately deterministic — engineers do not spend time identifying accountable parties	0 — None	1 — Partial	2 — Defined	3 — Governed	/3	*Score:*
SUBTOTAL — Ownership and Accountability					**/12**	*Total:*

Dimension 5 — Governance at the Point of Execution

Does governance operate before automation, pipelines, and AI agents act — or does your organization discover governance violations after they have already occurred?

Capability	0	1	2	3	Max	Score
Pre-execution change validation Changes are validated against architectural intent before pipelines, automation, or AI agents execute them	0 — None	1 — Partial	2 — Defined	3 — Governed	/3	*Score:*
Automated approval for conforming changes Changes that meet defined criteria are approved automatically — human review is reserved for genuinely ambiguous cases	0 — None	1 — Partial	2 — Defined	3 — Governed	/3	*Score:*
Deterministic violation blocking Changes that violate constraints are blocked automatically — not routed to human review as a default mechanism	0 — None	1 — Partial	2 — Defined	3 — Governed	/3	*Score:*
AI and automation bounds AI agents and autonomous systems operate within explicitly defined boundaries — not under inferred or assumed constraints	0 — None	1 — Partial	2 — Defined	3 — Governed	/3	*Score:*
SUBTOTAL — Governance at Execution					**/12**	*Total:*

Dimension 6 — Evidence and Auditability

Does your organization produce continuous, automatically generated evidence of infrastructure state, intent, and governance decisions — or is evidence assembled manually for audits and incidents?

Capability	0	1	2	3	Max	Score
Automated evidence generation Audit evidence is produced as a byproduct of operation — not assembled retrospectively from logs, tickets, and documentation	0 — None	1 — Partial	2 — Defined	3 — Governed	/3	*Score:*
Intent-to-action traceability Every infrastructure action can be traced to the	0 — None	1 — Partial	2 — Defined	3 — Governed	/3	*Score:*

Capability	0	1	2	3	Max	Score
intent or blueprint that authorized it						
Continuous compliance posture Compliance posture is a continuous, queryable state — not a point-in-time assessment	0 — None	1 — Partial	2 — Defined	3 — Governed	/3	*Score:*
Evidence durability and accessibility Evidence is retained in a queryable, structured form — not scattered across log archives and spreadsheets	0 — None	1 — Partial	2 — Defined	3 — Governed	/3	*Score:*
SUBTOTAL — Evidence and Auditability					**/12**	*Total:*

Maturity Assessment Summary

Dimension	Max	Your score	Priority
1 — Infrastructure State Modeling	/12	*Enter score:*	*High / Medium / Low:*
2 — Intent and Blueprint Governance	/12	*Enter score:*	*High / Medium / Low:*
3 — Continuous Reconciliation	/12	*Enter score:*	*High / Medium / Low:*
4 — Ownership and Accountability	/12	*Enter score:*	*High / Medium / Low:*
5 — Governance at the Point of Execution	/12	*Enter score:*	*High / Medium / Low:*
6 — Evidence and Auditability	/12	*Enter score:*	*High / Medium / Low:*
TOTAL SCORE	**/72**	*Total:*	

Score	Maturity band	What it means	Recommended starting point
0–18	**Pre-IOM**	Infrastructure governance is primarily manual and reactive. Significant foundational work is required before IOM adoption begins.	*Start with state modeling — get an accurate, automated view of what exists before addressing governance.*
19–36	**Emerging**	Some capabilities exist but are inconsistent, isolated, or tool-specific. The organization recognizes the need for a governing model but has not yet established one.	*Focus on intent definition and ownership encoding — these unlock the consistency that existing tools lack.*
37–54	**Developing**	Core IOM capabilities are present and partly operational. Gaps exist in reconciliation frequency, pre-execution validation, or cross-domain coverage.	*Close the reconciliation and pre-execution gaps — these convert existing capability into active governance.*
55–72	**Governed**	IOM capabilities are operating continuously and improving. The organization is ready to extend governance to AI agents and autonomous	*Focus on AI-readiness — establish authoritative context and explicit boundaries for autonomous operation.*

Score	Maturity band	What it means	Recommended starting point
		systems.	

PART 2

Tool Mapping Worksheet

For each IOM capability domain below, record the tools your organization currently uses and identify the gap between what those tools provide and what the IOM standard requires. This worksheet produces the input for Part 3 (Gap Analysis).

> **How to read this table** The 'IOM requirement' column describes what a fully conformant IOM must provide in this domain. Your task is to assess whether your current tooling satisfies that requirement — and if not, to characterize the gap. Be specific: note if the tool provides the capability partially, periodically, or only for a subset of your environments.

Capability domain	IOM requirement	Your current tools	Gap / notes
INFRASTRUCTURE STATE			
Asset and configuration inventory	All infrastructure assets discovered automatically and continuously — not human-curated. Records include configuration state, not just existence.	*Tool(s) currently used:*	*Gap / notes:*
Relationship and dependency modeling	Relationships between assets are explicitly modeled with directionality and semantics — not inferred from traffic or logs. Blast-radius boundaries are first-class properties.	*Tool(s) currently used:*	*Gap / notes:*
Cross-domain unified model	A single model covers cloud providers, on-premises, network, identity, and edge — not separate inventories per domain requiring manual correlation.	*Tool(s) currently used:*	*Gap / notes:*
INTENT AND BLUEPRINTS			
Architectural intent encoding	Infrastructure intent is encoded in a machine-readable, versioned format separate from IaC — not embedded in comments, wikis, or runbooks.	*Tool(s) currently used:*	*Gap / notes:*
Blueprint governance	Approved design patterns exist as reusable blueprints that govern validation across environments — not as IaC templates used solely for provisioning.	*Tool(s) currently used:*	*Gap / notes:*
Constraint definition and enforcement	Security, compliance, and architectural constraints are defined at the intent level and enforced automatically — not only checked in pipelines.	*Tool(s) currently used:*	*Gap / notes:*

Capability domain	IOM requirement	Your current tools	Gap / notes
CONTINUOUS RECONCILIATION			
Built vs. running state comparison	Designed state and actual running state are compared continuously. Deviations are classified by severity and domain impact — not just surfaced as alerts.	*Tool(s) currently used:*	*Gap / notes:*
Pre-execution validation	Changes are validated against intent before execution — not detected as violations after automation has already acted.	*Tool(s) currently used:*	*Gap / notes:*
Drift prevention vs. detection	The operating model prevents drift from occurring through pre-execution controls — not solely detects it after the fact through scanning.	*Tool(s) currently used:*	*Gap / notes:*
OWNERSHIP AND AUTHORITY			
Explicit ownership encoding	Every infrastructure element has a machine-readable, enforced owner — not a team tag in a CMDB that may be stale.	*Tool(s) currently used:*	*Gap / notes:*
Decision authority mapping	Who may authorize changes, exceptions, and escalations is explicitly modeled and enforced — not embedded in ticket routing conventions.	*Tool(s) currently used:*	*Gap / notes:*
Blast-radius awareness at decision time	The impact scope of any proposed change is deterministic before execution — not reconstructed after an incident.	*Tool(s) currently used:*	*Gap / notes:*
EVIDENCE AND AUDITABILITY			
Automated evidence generation	Audit evidence is produced continuously as a byproduct of governed operation — not assembled manually for compliance cycles.	*Tool(s) currently used:*	*Gap / notes:*
Intent-to-action traceability	Every infrastructure action is traceable to the blueprint or intent that authorized it — creating a durable chain from design to execution.	*Tool(s) currently used:*	*Gap / notes:*
Continuous compliance posture	Compliance posture is a live, queryable state derived from the reconciled model — not a periodic assessment against a point-in-time snapshot.	*Tool(s) currently used:*	*Gap / notes:*

PART 3

Gap Analysis

Translate your maturity scores and tool mapping observations into a prioritized gap inventory. For each gap, record the capability dimension, the specific gap, its severity, the effort required to close it, and the primary driver (compliance, AI readiness, operational reliability, cost control, security).

Gap prioritization guidance

- **Severity HIGH:** Gap creates active risk — security exposure, compliance failure, or AI operating without defined boundaries. Address before any other gaps.
- **Severity MEDIUM:** Gap creates operational inefficiency or limits your ability to achieve a defined organizational goal. Address in Phase 2.
- **Severity LOW:** Gap represents an improvement opportunity but does not create active risk or block organizational goals. Address in Phase 3–4.

#	Capability dimension	Specific gap description	Severity	Effort	Primary driver	Phase
1	*Dimension:*	*Describe the gap:*	*H/M/L*	*S/M/L*	*Driver:*	*1/2/3/4*
2	*Dimension:*	*Describe the gap:*	*H/M/L*	*S/M/L*	*Driver:*	*1/2/3/4*
3	*Dimension:*	*Describe the gap:*	*H/M/L*	*S/M/L*	*Driver:*	*1/2/3/4*
4	*Dimension:*	*Describe the gap:*	*H/M/L*	*S/M/L*	*Driver:*	*1/2/3/4*
5	*Dimension:*	*Describe the gap:*	*H/M/L*	*S/M/L*	*Driver:*	*1/2/3/4*
6	*Dimension:*	*Describe the gap:*	*H/M/L*	*S/M/L*	*Driver:*	*1/2/3/4*
7	*Dimension:*	*Describe the gap:*	*H/M/L*	*S/M/L*	*Driver:*	*1/2/3/4*
8	*Dimension:*	*Describe the gap:*	*H/M/L*	*S/M/L*	*Driver:*	*1/2/3/4*
9	*Dimension:*	*Describe the gap:*	*H/M/L*	*S/M/L*	*Driver:*	*1/2/3/4*
10	*Dimension:*	*Describe the gap:*	*H/M/L*	*S/M/L*	*Driver:*	*1/2/3/4*

Gap Analysis Notes

PART 4

Adoption Roadmap Planner

IOM adoption follows a proven four-phase progression. Each phase builds on the last and introduces no execution risk until authority has been established through accuracy. Use this planner to assign timelines, owners, and success criteria for each phase based on your organization's priorities and gaps.

> **The governing principle of IOM adoption** Do not introduce execution authority before modeling authority is established. Organizations that attempt to govern before they can accurately model their infrastructure produce unreliable decisions that erode trust in the operating model. Phase 1 is not optional — it is the foundation everything else depends on.

Phase	Goal	Key activities	Success criteria	Your timeline / notes
Phase 1 Model	Establish an accurate, continuously updated canonical model of your infrastructure — read-only. No	Ingest target environments into the canonical model. Normalize relationships and dependencies. Identify undocumented assets, shadow infrastructure, and ownership	*Proposed (draft) model covers target environments with* >95% asset accuracy. Ownership gaps identified and surfaced. Drift baseline established. No production systems modified.	*Your timeline / notes:*

Phase	Goal	Key activities	Success criteria	Your timeline / notes
	governance authority yet.	gaps. Baseline drift frequency and type.		
Phase 2 Reconcile and govern	Activate continuous reconciliation and establish governance against declared intent. Begin enforcing architectural constraints.	Define intent and blueprints for target environments. Activate continuous built-vs-running-state reconciliation. Apply governance rules: auto-approve conforming, block violations, escalate ambiguous. Establish ownership and blast-radius semantics.	*Governance operating pre-execution for defined environments. Conforming changes auto-approved without human review. Violations blocked deterministically. Ownership explicitly encoded and enforced.*	*Your timeline / notes:*
Phase 3 Govern automation	Extend IOM authority into IaC pipelines, CI/CD, and automation systems. Human escalation becomes the exception, not the default.	Integrate IOM validation into IaC and CI/CD pipelines. Bind automation frameworks to governance boundaries. Establish escalation paths for genuinely ambiguous changes. Expand scope to additional environments and domains.	*IaC changes validated against IOM before execution across all governed environments. Human escalation rate* **<**15% of all changes. Change cycle time reduced by target percentage. No production violations in governed environments.	*Your timeline / notes:*
Phase 4 Enable AI operations	Extend authoritative context and governed boundaries to AI agents and autonomous systems. Achieve explainable, auditable autonomy.	Provide authoritative infrastructure context to AI systems. Define explicit bounds for autonomous action. Establish AI decision audit trails traceable to intent. Activate learning loops to refine blueprints from operational evidence.	*AI agents operating within governed boundaries with full audit trail. All autonomous actions traceable to authoritative intent. Compliance posture continuously queryable without manual effort. Governance intelligence improving through operational evidence.*	*Your timeline / notes:*

Roadmap Planning Worksheet

Phase	Target environment / scope	Target start date	Target completion	Owner / responsible team
Phase 1 — Model	*Environment:*	*Date:*	*Date:*	*Owner:*
Phase 2 — Reconcile and govern	*Environment:*	*Date:*	*Date:*	*Owner:*
Phase 3 — Govern automation	*Environment:*	*Date:*	*Date:*	*Owner:*
Phase 4 — Enable AI operations	*Environment:*	*Date:*	*Date:*	*Owner:*

PART 5

IOM Conformance Checklist

Use this checklist to evaluate any solution — commercial or open source — claiming to provide Infrastructure Operating Model capabilities. The checklist is derived directly from the IOM Standard v1.0

normative requirements. A solution that cannot satisfy all mandatory requirements does not constitute a conformant IOM.

> **How to use this checklist** For each requirement, ask the vendor or project to demonstrate — not describe — the capability. Demonstrations should use your infrastructure data, not synthetic examples. A capability that exists in a roadmap or beta does not satisfy a mandatory requirement.

Mandatory requirements — all must be satisfied

	Requirement (IOM Standard v1.0 normative)	How to verify	Evaluation notes
[ ]	**Authoritative state modeling** The solution maintains a continuously updated, canonical model of infrastructure state — including assets, relationships, configurations, and dependencies — without relying on human-maintained records or periodic snapshots.	*Ask: Show me how the model is updated when a new service is deployed. How quickly does the model reflect reality? How are relationships captured?*	*Notes:*
[ ]	**Explicit intent and blueprint governance** Infrastructure intent is encoded as explicit, machine-readable blueprints — separate from IaC or configuration code. Intent is declared, not inferred from runtime behavior.	*Ask: Show me how intent is defined. Where does it live? How is it versioned? Show me a blueprint and explain how it differs from a Terraform module.*	*Notes:*
[ ]	**Continuous built-vs-running-state reconciliation** The solution continuously compares intended state, built state, and actual running state. Reconciliation is real-time — not a scheduled scan.	*Ask: Show me the reconciliation running live. How is drift classified? What happens when running state diverges from built state?*	*Notes:*
[ ]	**Pre-execution validation** Governance decisions are made before execution — not after. Proposed changes are validated against intent and constraints before automation, pipelines, or AI agents act.	*Ask: Show me a change being validated before it executes. What happens when a change violates a blueprint constraint? Can a violating change proceed?*	*Notes:*
[ ]	**Explicit ownership and blast-radius awareness** Ownership of every infrastructure element is explicitly encoded and machine-readable. Blast radius for any change is deterministic before execution — not estimated.	*Ask: Show me who owns this service. Show me what the blast radius of this change is. How is ownership kept current?*	*Notes:*
[ ]	**Automated evidence generation** Audit evidence is produced	*Ask: Show me the evidence trail for a recent change. How would I answer an auditor*'s question about what changed and who	*Notes:*

	Requirement (IOM Standard v1.0 normative)	How to verify	Evaluation notes
	continuously as a byproduct of governed operation. Evidence is structured, queryable, and traceable from intent to action.	authorized it?	
[ ]	**AI-safe reasoning substrate** AI systems operating on top of the platform consume authoritative infrastructure context and explicit intent boundaries — not inferred signals. AI decisions are bounded, explainable, and auditable.	*Ask: Show me how an AI agent*'s action is bounded by the operating model. What prevents an AI agent from acting outside defined intent? Show me the audit trail for an AI-initiated action.	*Notes:*

Claims that do not satisfy IOM requirements

Vendor claim	Why it does not satisfy IOM requirements	The question to ask
"We provide a digital twin"	Digital twins describe infrastructure through observation. They are outside the execution path and advisory in function. An IOM governs through authority — it sits upstream of execution and makes binding determinations of legitimacy.	*Does your twin sit in the execution path as an authority layer? Or does it observe after the fact?*
"We manage infrastructure state"	State management describes what exists. An IOM continuously reconciles what exists against what is intended to exist — and enforces that relationship. State management without reconciliation is an inventory, not a governing model.	*Is your state continuously reconciled against declared intent? Or is it updated to reflect what was deployed?*
"We enforce policy"	Policy enforcement evaluates rules. An IOM evaluates rules in the context of an authoritative, continuously reconciled model of infrastructure state, intent, and ownership. Rules without architectural context cannot determine legitimacy or blast radius.	*Are your policies evaluated against an authoritative infrastructure model? Or are they rule-checked in isolation?*
"We have continuous compliance"	Continuous compliance scanning detects violations after they occur. An IOM prevents violations from occurring through pre-execution validation — compliance is a byproduct, not a scan result.	*Does compliance happen before execution — or do you detect non-compliance after it occurs?*
"We support AI operations"	Supporting AI with telemetry or recommendations is not the same as providing an authoritative substrate. AI must consume explicitly defined intent and boundaries — not infer them from logs, metrics, or patterns.	*Does your AI operate on authoritative, pre-validated context? Or does it infer intent from operational data?*

Vendor claim	Why it does not satisfy IOM requirements	The question to ask
"We do continuous discovery"	Discovery builds an inventory of what exists. It does not encode intent, enforce constraints, or validate actions before execution. An IOM requires all three — discovery is a necessary input, not an IOM in itself.	*Is your discovery model used to validate actions before execution? Or is it a visibility tool?*

Next steps

After completing this kit

- **1. Score interpretation:** Review your maturity scores as a team. Scores below 2 in Dimensions 3 (Reconciliation) or 5 (Governance at Execution) indicate the highest-priority gaps — these are the capabilities that define the difference between an IOM and a collection of governance tools.
- **2. Gap prioritization:** Identify your top three gaps from Part 3. For each, determine whether the gap is addressable through configuration and process changes to existing tools, or whether it represents a structural absence requiring a new capability.
- **3. Phase 1 scoping:** Select an initial environment for IOM Phase 1 adoption. Regulated environments, high-change environments, or environments where AI operations are planned are the highest-value starting points. The initial scope should be small enough to complete modeling in 30–60 days.
- **4. Working group:** Consider joining the IOM working group at theIOM.org. Practitioners who have completed this assessment and are pursuing adoption are eligible for working group membership and contribute directly to the evolution of the standard.

theIOM.org | IOM Standard v1.0 | Open Standard | Community Governed

Infrastructure Operating Model Standard | Open Standard | Community Governed Page