AI-Accelerated UI/UX Transformation Playbook:

A Practical Guide for Modernizing Dashboards with AI

Author: Ross Mallory

Date: 03/04/2026

Created in Partnership with Strategic AI Program Advisors

AI Accelerated UI/UX Transformation Playbook: A Practical Guide for Modernizing Dashboards with AI

Contents

1. Purpose of This Playbook

This playbook delivers a **clear, repeatable framework** for modernizing legacy UI and UX experiences using AI-assisted tools. Cross-functional teams, comprising analysts, designers, developers, and business stakeholders, will receive practical guidance relevant to each stage of the transformation process.

Leaders can expect a structured, end-to-end methodology that enables consistent results and accelerates modernization. The framework emphasizes practical execution, not theoretical concepts.

The playbook addresses four essential outcomes:

- Teams reduce manual and repetitive tasks through automation.
- Project leaders accelerate iteration cycles to deliver value faster.
- Designers and stakeholders improve clarity and usability of dashboards for better decision-making.
- Governance processes and human oversight remain strong throughout modernization efforts.

2. How to Use This Playbook

Ways to Use This Playbook

This playbook is designed to be flexible and actionable, supporting teams with different needs and levels of experience. There are three primary ways to leverage its guidance:

- **Facilitated Workshop** — Conduct a structured workshop over one or two days. This approach is ideal for cross-functional groups aiming to make rapid progress on UI/UX modernization. Facilitators can guide teams through each play, encourage hands-on engagement, and foster collaborative problem-solving. Sample prompts and step-by-step instructions make it easy to get started, drive alignment, and surface actionable insights.
- **Self-Paced Pilot Guide** — Follow the playbook as a pilot guide for a small product team. This format supports incremental adoption, allowing teams to experiment with AI-assisted workflows, iterate on design solutions, and gradually integrate best practices into daily routines. Each play is designed with clear objectives, inputs, and validation criteria so teams can monitor progress and measure outcomes independently.
- **Reference Manual** — Utilize the playbook as a comprehensive manual to repeat and scale successful patterns. Teams can reference the playbook to support onboarding, reinforce standards across projects, and troubleshoot common challenges. The manual includes reusable templates, prompt logs, and practical examples to streamline future efforts and build institutional knowledge.

Regardless of how you use the playbook, the information offers a structured, step-by-step framework that can be tailored to fit the scope and goals of your modernization initiative.

Structure and Content of Each "Play"

To ensure clarity and repeatability, every play in this playbook follows a consistent structure. This enables teams to quickly understand the purpose of each play, gather necessary materials, execute confidently, and validate results.

The elements include:

- **Objective** — Defines what the play aims to achieve, aligned with modernization goals.

- **Inputs and Materials** — Lists required data, tools, and resources needed to initiate the play.
- **Execution Steps** — Outlines step-by-step actions for carrying out the play, including collaboration guidance and checkpoints.
- **Example AI Prompts** — Provides sample prompts to maximize the effectiveness of AI-assisted tools and workflows.
- **Expected Outputs** — Describes the tangible results and deliverables for each play, ensuring teams can track progress and demonstrate value.
- **Validation Criteria** — Specifies how to assess whether the outputs meet the desired standards and objectives, supporting continuous improvement.

This consistent structure helps teams systematically modernize legacy UI/UX, fosters accountability, and makes it easy to adapt the playbook to future projects.

Playbook Tip: Capture every prompt and AI output in the Prompt Log. Maintaining a comprehensive log builds an internal library of reusable patterns, accelerates learning, and enables teams to refine their approach based on past successes and lessons learned.

3. Case Study Context and Applicability

Reference Case Study

This playbook uses a Scorecard modernization effort as a running example.

In that effort:

- A legacy Microsoft Access backend was replaced with a web-ready architecture
- A complex Scorecard UI was redesigned for clarity and speed

This playbook further shows how AI tools could be used now to accelerate analysis, design, and development.

Applicability Beyond the Case Study

While examples reference a Scorecard, the plays are **intentionally generic** and apply to:

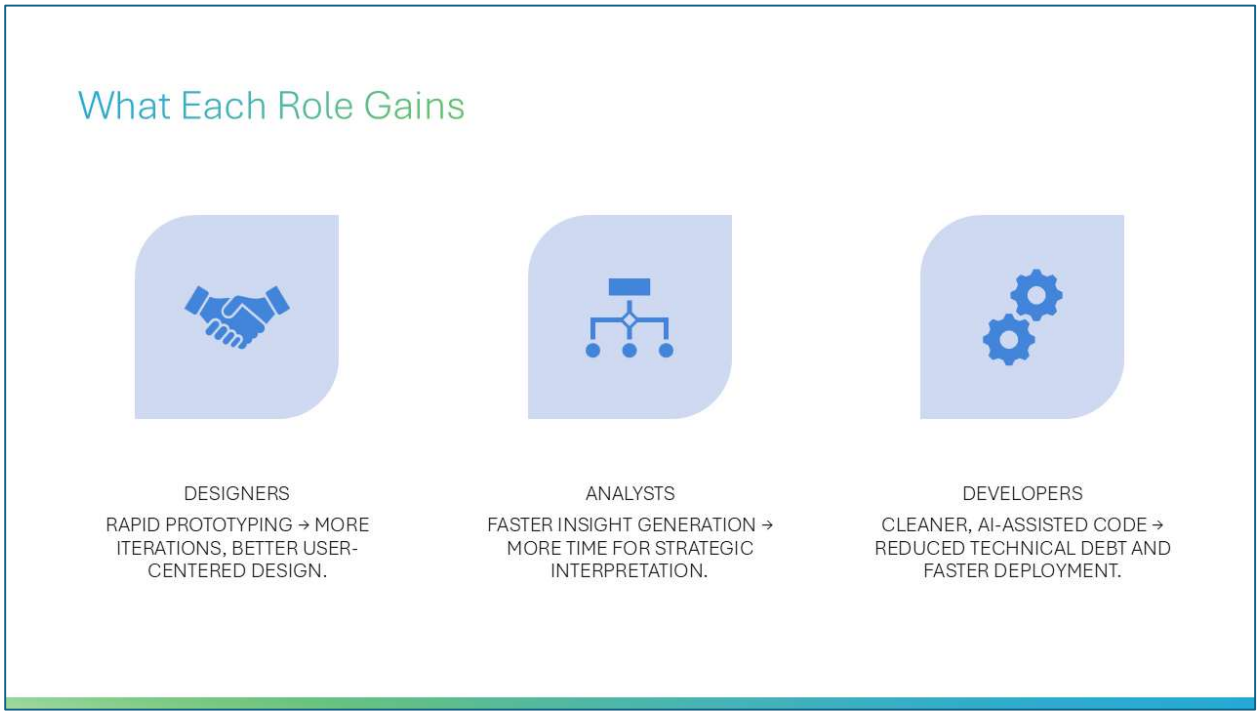
- Operational dashboards
- Executive reporting tools
- Monitoring and compliance systems
- Internal analytics products

4. The AI-Accelerated UI/UX Workflow

This playbook follows a **consistent workflow** that can be reused across pilots.

Workflow Overview

1. Ingest and clean data into a web-ready backend
2. Generate AI-assisted analysis and summaries
3. Produce wireframe layouts from natural language prompts
4. Validate layouts using attention analysis
5. Scaffold reusable UI components with AI
6. Review, refine, and deploy iteratively
7. Measure outcomes and capture learnings



5. Play 1: AI-Assisted Data Design

Objective

Use AI tools to quickly generate a **clear narrative summary** and a **recommended visualization** from structured data.

Inputs

- Cleaned dataset (CSV or table)
- Business question or decision context
- Access to Microsoft Copilot or equivalent

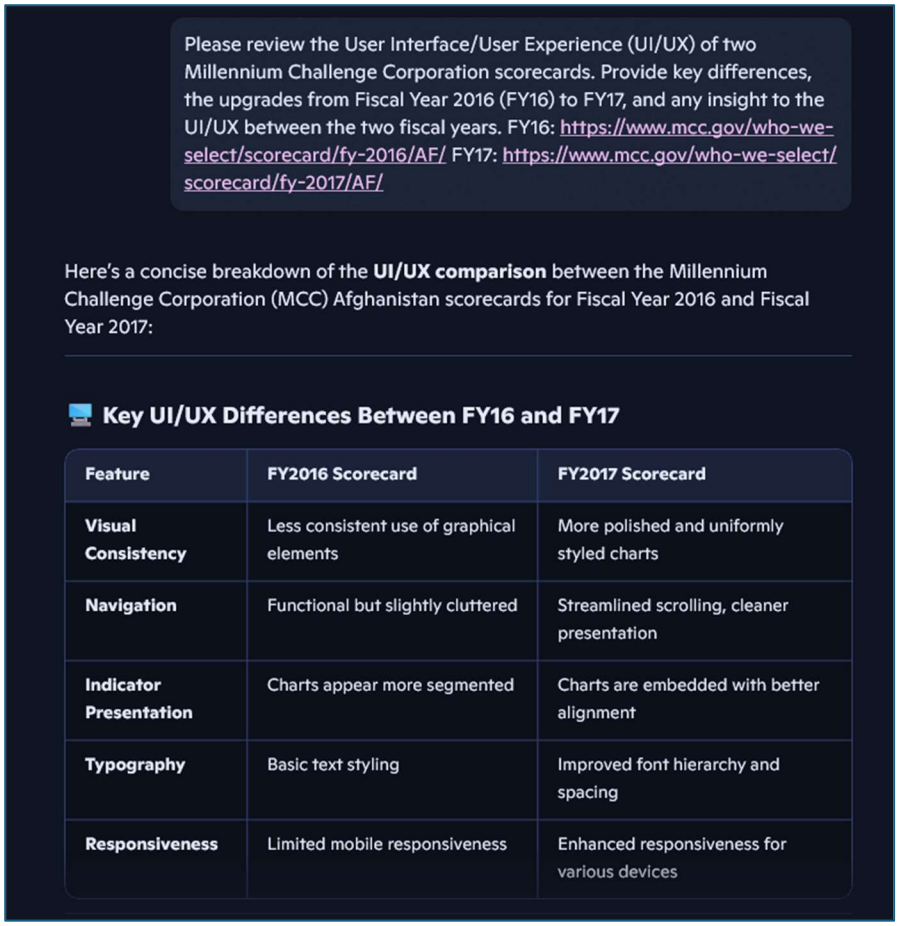
Execution Steps

1. Define the indicator and scope
2. Prompt AI to summarize trends and drivers
3. Review the generated narrative aloud
4. Validate numbers against the source data
5. Refine language for non-technical stakeholders
6. Select and document the visualization choice

Example Prompts

These prompts should be updated for your case study or project. The below prompts can be tailored to your work.

- “Summarize data for [country] in [indicator] using [column].”
- “Identify the top three ways to improve user interaction with the Scorecards new upload tool. (*note: this is the backend tool to generate the Scorecards)
- “Recommend flow updates so the user sees the Scorecard information while the calculations are being processed.”



*Plain language prompt example

Expected Outputs

- Two to three sentence narrative summary
- Recommended chart type updates with rationale
- List of follow-up questions or gaps

Validation Criteria

- Numerical accuracy
- Plain-language clarity
- Actionability for decision makers

6. Play 2: AI-Generated Wireframes and Attention Validation

Objective

Rapidly generate dashboard wireframes using AI and validate visual focus using attention-mapping tools.

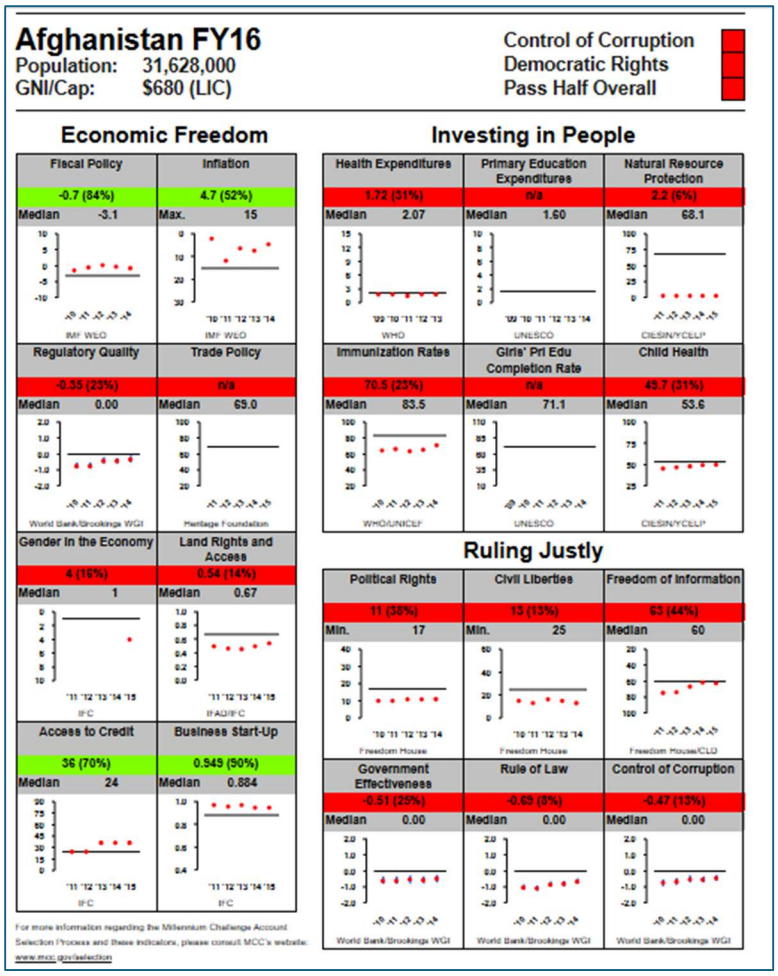
Inputs

- **Use your case description:** Clearly define the scenario or business problem the dashboard aims to address, including intended outcomes and target audience. Specify the context in which the dashboard will be used (e.g., operational monitoring, strategic review, or performance tracking).
- **Top three user tasks:** List the most critical actions users must accomplish using the dashboard (e.g., identify trends, monitor thresholds, or spot anomalies). For each task, describe the workflow and any supporting information required to enable efficient completion. The more context the better. This information can also be added iteratively, starting with a base set of information to receive the first results. Then add more context and ask for a revised version. Then done as many times as needed.
- **Key metrics and thresholds:** Provide a comprehensive list of quantitative indicators that will be featured, including definitions, units, and target values or alert thresholds. Specify how these metrics relate to business objectives and how they should be prioritized visually within the dashboard. Starting with key metrics is important. The metrics are where human-in-the loop get utilized. Create the first metric and then confirm, then add more metrics or a single metric, and iterate.

Execution Steps

1. **Prompt AI to generate a dashboard layout:** Formulate a clear prompt to the AI tool, specifying layout priorities, visual hierarchy, and required components. Provide sample data or sketches if available to improve output relevance. Review the initial wireframe for completeness and alignment with user tasks and metrics. This again is an area where focusing on one step of the dashboard at a time may be more practical and not cause for long delays, with each step being added iteratively upon confirmation of the previous step for accuracy and consistency.
2. **Export the wireframe or mockup:** Save the generated layout in a format suitable for collaboration and further analysis (e.g., PNG, PDF, or interactive prototype). Ensure all dashboard elements are clearly labeled and documented for stakeholders.

3. **Run attention analysis on the layout:** Use attention-mapping tools to assess which elements attract the most visual focus. Analyze heatmaps or gaze plots to verify that key metrics and actionable items are prioritized.
4. **Identify low-attention risk areas:** Review results to pinpoint dashboard zones that are overlooked or under-emphasized. Document any gaps where important information may not be noticed by users.
5. **Adjust hierarchy, contrast, and spacing:** Refine the wireframe by modifying visual properties such as color, font weight, size, and placement to increase the prominence of critical metrics and tasks. Consider accessibility guidelines and ensure the design is usable for all audiences.
6. **Re-run validation until priorities align:** Iterate attention analysis and wireframe adjustments until the layout consistently directs user focus to priority areas. Document the rationale for final design decisions and prepare the dashboard for stakeholder review or implementation.



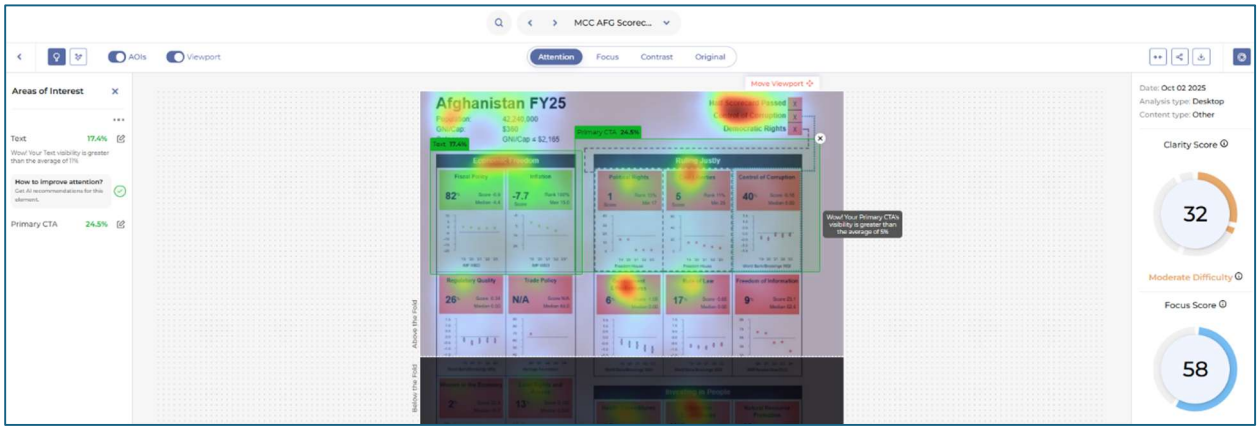
*Previous Millennium Challenge Corporation Scorecard FY16

Example Prompts

- “Create a dashboard layout that prioritizes thresholds and trends.”
- “List visual elements in order of importance and explain why.”

Expected Outputs

- Wireframe draft
- Attention heatmap
- Rationale for layout decisions



*New Scorecard heatmap

Playbook Tip: For each step, maintain clear documentation and rationale to support validation criteria: numerical accuracy, plain-language clarity, and actionability for decision makers. Engage stakeholders early and often to ensure the dashboard meets real-world needs and usability standards.

7. Play 3: AI-Assisted Developer Code Scaffolding

Objective

Use AI to scaffold reusable UI components while maintaining engineering standards.

Inputs

This approach establishes clarity and minimizes risks throughout the development process.

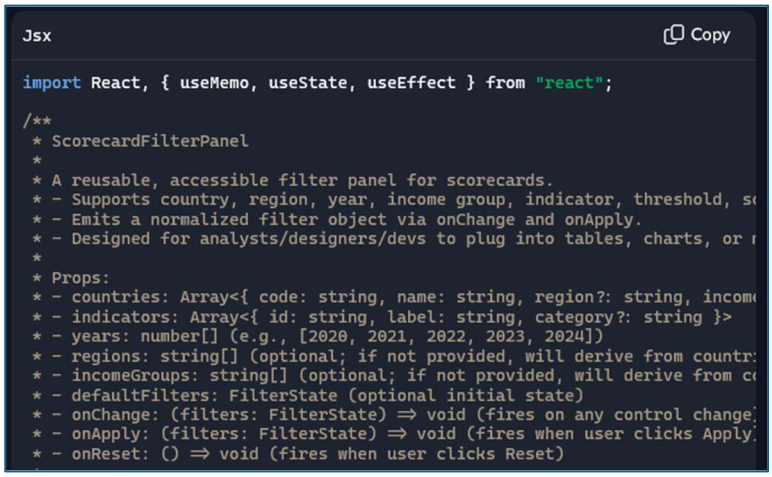
- **Component requirements:** Define the functional scope, intended use cases, and necessary props or data inputs. Specify any dependencies or integration points with other system components.
- **Accessibility and performance constraints:** Outline standards for accessibility (such as WCAG guidelines), responsive design requirements, and performance benchmarks that must be met.
- **Developer environment with GitHub Copilot:** Confirm that the development workspace is configured for AI-powered code generation, including IDE setup, repository structure, and access permissions.
- **Security and compliance requirements:** Identify any regulatory, privacy, or security standards that must be adhered to during development (e.g., GDPR, OWASP).

Execution Steps

Follow this structured workflow to maximize efficiency and maintain high standards when scaffolding UI components with AI:

1. **Define the component contract:** Document the API, expected behaviors, and interface requirements. Include diagrams or wireframes if possible. Validate requirements with stakeholders before proceeding.
2. **Generate scaffolded code with AI:** Use GitHub Copilot or similar AI tools to produce the initial codebase. Prompt the AI with detailed instructions and review suggestions for accuracy and completeness.
3. **Run the component in a sandbox:** Deploy the generated code in an isolated environment for early testing. Verify that basic functionality meets requirements and identify areas for improvement.
4. **Execute basic tests:** Implement unit and integration tests to confirm component reliability. Use automated testing frameworks and ensure coverage for edge cases.

5. **Perform structured code review:** Conduct peer reviews focusing on code quality, adherence to standards, and maintainability. Leverage code review logs and feedback tools to track changes and decisions.
6. **Refactor and document the final version:** Optimize the code for readability and performance. Add comprehensive documentation, including usage examples, troubleshooting tips, and links to related resources.
7. **Release and monitor:** Deploy the completed component to production or staging environments. Set up monitoring and analytics to track usage and capture feedback for future improvements.



Example Prompts

- "Generate a React component named [ComponentName] with props for [X]."
- "Create default values and a simple unit test."

Expected Outputs

- Component scaffold
- Unit test or Storybook entry
- Code review log

Playbook Tip: Leaders should establish clear requirements and validate these requirements with stakeholders, including the internal team before engaging AI tools to scaffold code.

8. Facilitator and Team Operating Model

Recommended Roles

- Facilitator
- Analyst
- Designer
- Developer

Group Size

- Optimal: 6–12 participants
- Rotate roles across plays to build shared understanding

Clearly defined roles ensure effective collaboration and accountability within the team. Below are the core recommended roles for successful project execution, along with brief descriptions of their primary responsibilities:

- **Facilitator:** Guides the workflow, keeps discussions on track, ensures inclusive participation, and resolves conflicts. The facilitator maintains momentum and ensures that objectives are met during each session.
- **Analyst:** Gathers requirements, interprets data, identifies opportunities and risks, and provides insights that inform decision-making. The analyst ensures the team's actions are grounded in evidence and user needs.
- **Designer:** Shapes the users' experience, creates wireframes or prototypes, and ensures solutions are accessible and intuitive. The designer collaborates closely with analysts and developers to translate requirements into impactful visuals and interactions.
- **Developer:** Implements and tests technical solutions, ensuring code quality and scalability. Developers collaborate with designers to bring concepts to life and with analysts to validate technical feasibility.

Depending on project complexity, additional roles may be assigned (e.g., Product Owner, QA Lead, or Stakeholder Liaison). Each team member should understand their responsibilities and how they contribute to the overall objectives.

Group Size

Selecting the appropriate group size and structure maximizes productivity and fosters a collaborative environment:

- **Optimal: 6–12 participants** — This range ensures enough diversity of perspectives while remaining small enough for effective communication and agile decision-making.
- **Role Rotation:** To build shared understanding and prevent silos, rotate key roles (such as facilitator, designer, and analyst) across sessions or project phases. This approach encourages cross-training, increases empathy for other roles, and promotes team resilience.

Playbook Tip: For larger initiatives, consider forming subgroups with dedicated facilitators who coordinate at regular intervals. Maintain clear communication channels and shared documentation to ensure alignment across the

9. Governance, Review, and Transparency

Core Principles

The foundation of governance, review, and transparency relies on clear, actionable principles that guide team behavior and decision-making throughout AI-enabled projects. Adhering to the following principles ensures trustworthy, auditable, and responsible outcomes:

- **Human validation for all AI outputs:** Every output generated by AI must be reviewed and verified by a designated human team member before further use or publication. This step safeguards against errors, bias, or unintended consequences and upholds accountability.
- **Prompt and output logging:** All prompts submitted to AI systems, along with their resulting outputs, should be systematically logged and stored. Maintaining comprehensive records provides traceability for audits, troubleshooting, and continuous improvement.
- **Clear data provenance:** Every dataset and information source used for AI processing must be documented, with explicit references to origin, collection date, and data quality checks. This transparency facilitates responsible data handling and enables root-cause analysis in case of issues.
- **Explicit approval checkpoints:** Major project milestones or deliverables require formal sign-off by relevant stakeholders, such as project leads or subject matter experts. Approval checkpoints ensure alignment with business objectives and regulatory requirements before proceeding.

Public-Facing Outputs

All outputs intended for external audiences must adhere to enhanced transparency standards.

To build trust and foster understanding, follow these steps for every public-facing deliverable:

Label outputs as AI-assisted: Clearly indicate when content, recommendations, or analyses have been generated or augmented by AI tools. This disclosure should be prominent and unambiguous.

Include a concise methodology note: Provide a brief description outlining the process used to generate the output, the tools involved, and the validation workflow.

At minimum, address the following elements:

- **Data sources:** Specify where the data originated, the scope, and any relevant preprocessing steps.
- **Validation steps:** Summarize how the output was checked for accuracy, consistency, and compliance with requirements, including human review and automated checks.
- **Reviewers:** List the names or roles of individuals responsible for final approval and oversight.

Playbook Tip: Maintain an accessible repository of public-facing outputs with their methodology notes, ensuring stakeholders and end users can easily verify the integrity and origins of published materials.

10. Measuring Pilot Success

*Track both **efficiency and quality** metrics:*

- Time per iteration
- Usability scores
- Feature adoption
- Analyst hours saved
- Qualitative user feedback

To effectively measure pilot success, you need to monitor and analyze a balanced set of efficiency and quality metrics. These provide a comprehensive view of performance, adoption, and user satisfaction, enabling data-driven decisions and ongoing improvement. Below are key metrics to consider, along with recommended methods for collection and interpretation:

- **Time per iteration:** Record the duration required to complete each workflow or cycle. Compare baseline (pre-pilot) and post-pilot timings to assess improvements. Aim to identify bottlenecks and optimize process steps for greater speed.
- **Usability scores:** Gather user ratings through surveys or standardized tools (e.g., System Usability Scale). Track trends over time and segment by user role or experience level to pinpoint areas requiring user interface enhancements.
- **Feature adoption:** Monitor which features are being used, frequency of usage, and the depth of engagement. Use analytics dashboards to visualize adoption rates and identify underutilized capabilities for targeted training or communication.
- **Analyst hours saved:** Calculate the reduction in manual work or repetitive tasks for analysts. Compare historical workload logs to pilot results and validate with direct feedback to ensure reported savings reflect actual experiences.
- **Qualitative user feedback:** Conduct interviews, focus groups, or open-ended survey questions to capture nuanced insights. Analyze feedback for recurring themes, pain points, and suggestions for improvement, prioritizing actionable recommendations.

Playbook Tip: Consider using dashboards for real-time visibility. Set targets or thresholds to guide decision-making.

11. Scaling and Next Steps

Immediate Actions

- Establish governance and logging
- Select a focused pilot
- Run plays end-to-end
- Capture prompts and artifacts

Longer-Term Actions

- Scale reusable components
- Maintain a metrics dashboard
- Build a community of practice