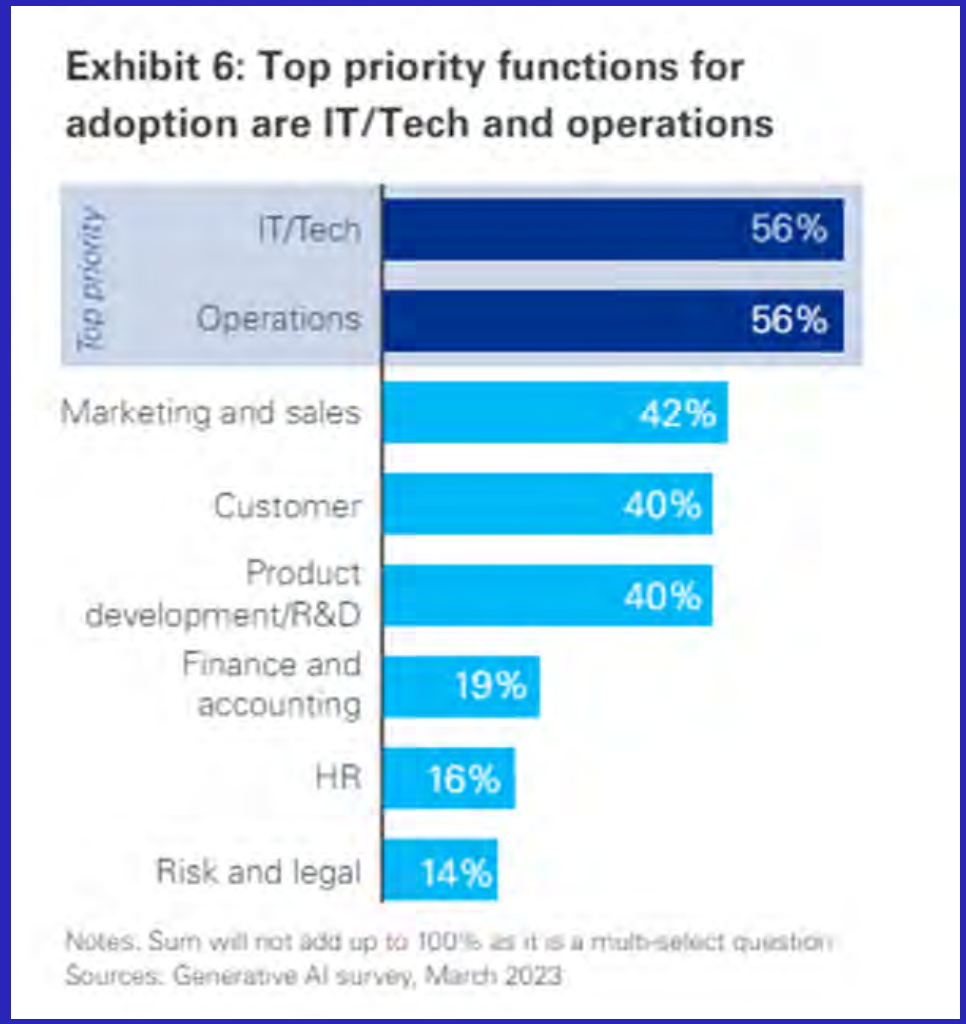
77% Expects Generative AI to have the largest impact on their business out of all emerging technologies

73% Believe Generative AI will increase workforce productivity

71% Will implement their first Generative AI solution within next two years

64% Believe Generative AI will help their business gain a competitive advantage over competitors

KPMG Generative AI Survey 2023



KPMG Generative AI Survey 2023

SRE 2.0 : Amplifying Reliability with GenAI

Indika Wimalasuriya

SRE - 2024

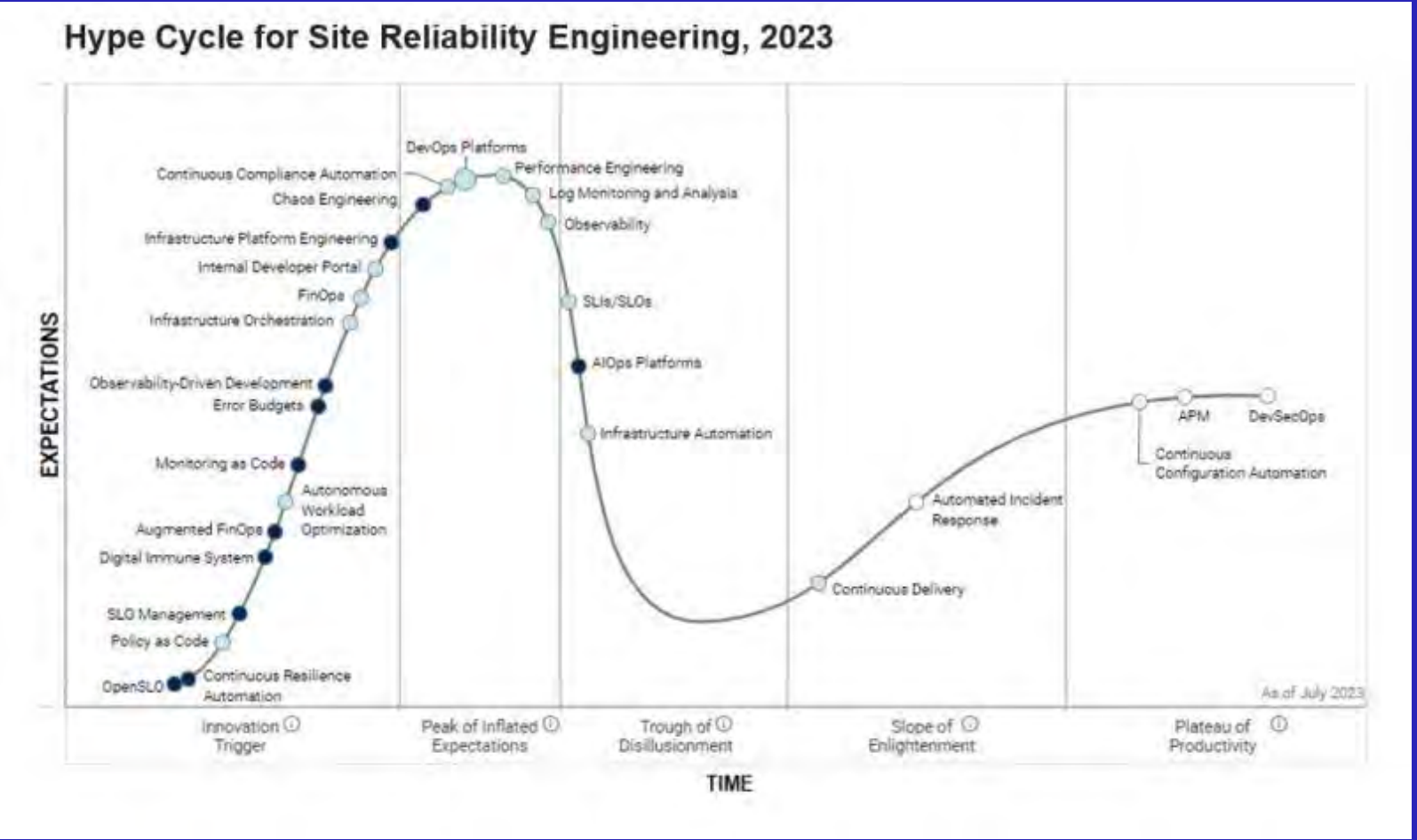
Agenda

- Challenges in Traditional SRE.
- The Role of GenAI in SRE
- GenAI Impacts in Key Pillars of SRE
- GenAI Use Cases and Potential Benefits
- Implementation Strategies
- Best Practices and Pitfalls to Avoid
- Future Outlook

Quick Intro about myself

- Resides in Colombo, Sri Lanka, with my beautiful daughter and wife.
- Reliability Engineering Advocate, Solution Architect (specializing in SRE, Observability, AIOps, & GenAI).
- Employed at Virtusa, overseeing technical delivery and capability development.
- Passionate Technical Trainer.
- Energetic Technical Blogger.
- AWS Community Builder - Cloud Operations.
- Ambassador at DevOps Institute (PeopleCert).

Gartner SRE Hype Cycle



- Focus on trajectory of emerging technologies and frameworks for SRE practices
- Aims to ensure reliability, maximize performance, and minimize resources

Site Reliability Engineering

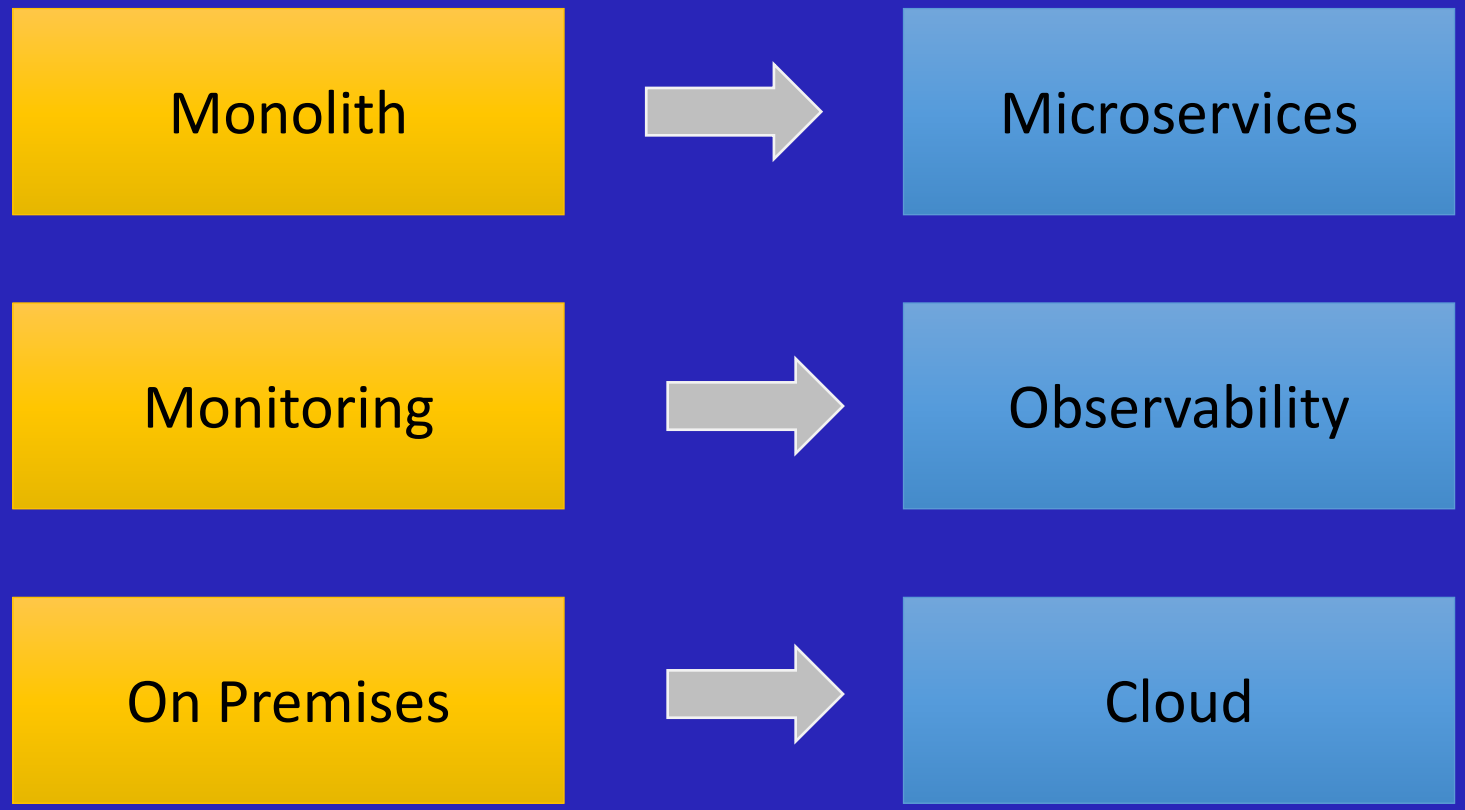
Key pillars of Success

- Reduce organizational Silos
- Accept failures as normal
- Implement gradual changes
- Leverage tooling and automating
- Measure everything

Principles

- Observability
- SLI, SLO and Error Budgets
- System architecture and Recovery Objectives
- Release & Incident Engineering
- Automation
- Resilience Engineering
- Blameless Postmortems

Navigating Digital Transformation: Managing Ever-Growing Complexity



- **Expansion of Data Sources**
- **Surge in Data Volume**
- **Exponential Rise in Failure Scenarios**

Operations is a Software Problem

By 2026,

Code written by developers/humans will be reduced by 50% due to generative AI code generation models

GenAI Emerges: Unveiling the Power of Next-Gen Artificial Intelligence

Generative AI, refers to machine learning models that can create new, original content like text, images, audio and more

Text Generation:
- Articles & stories - GPT-3 can generate news articles and fiction stories
- Code - GitHub Copilot suggests context-relevant code for developers

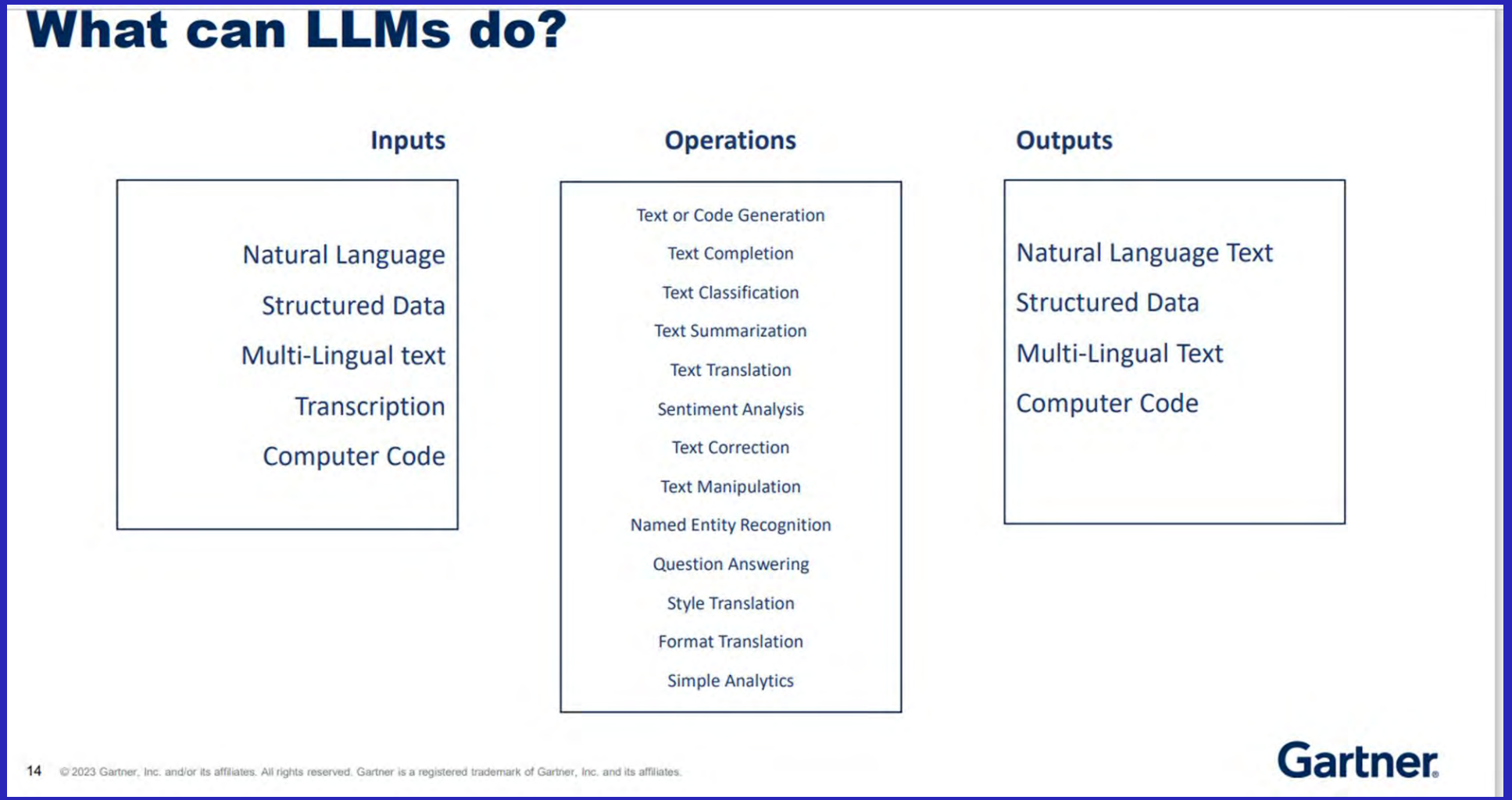
Image Generation:
- Art - DALL-E 2 creates original digital art from text prompts
- Photos - Generative models can edit or enhance photos

Video Generation:
- AI models can generate or edit video content

Drug Discovery:
- Models can analyze chemical compounds and suggest new drug candidates

Other Creative Applications:
- Generating logos, recipes, fashion designs, architectural drawings
- Personalizing content like customized ads or product recommendations

Unveiling the Potential: The Capabilities of LLM (Large Language Models)



Navigating Challenges: Risks Associated with Large Language Models

Model Risks

- Bias
- Misinformation
- Privacy
- Lack of context
- Lack of creativity
- Lack of Explainability

Misuse Risks

- Misinformation
- Cyberbullying
- Phishing
- Automated Generation
- Fraud
- Malware

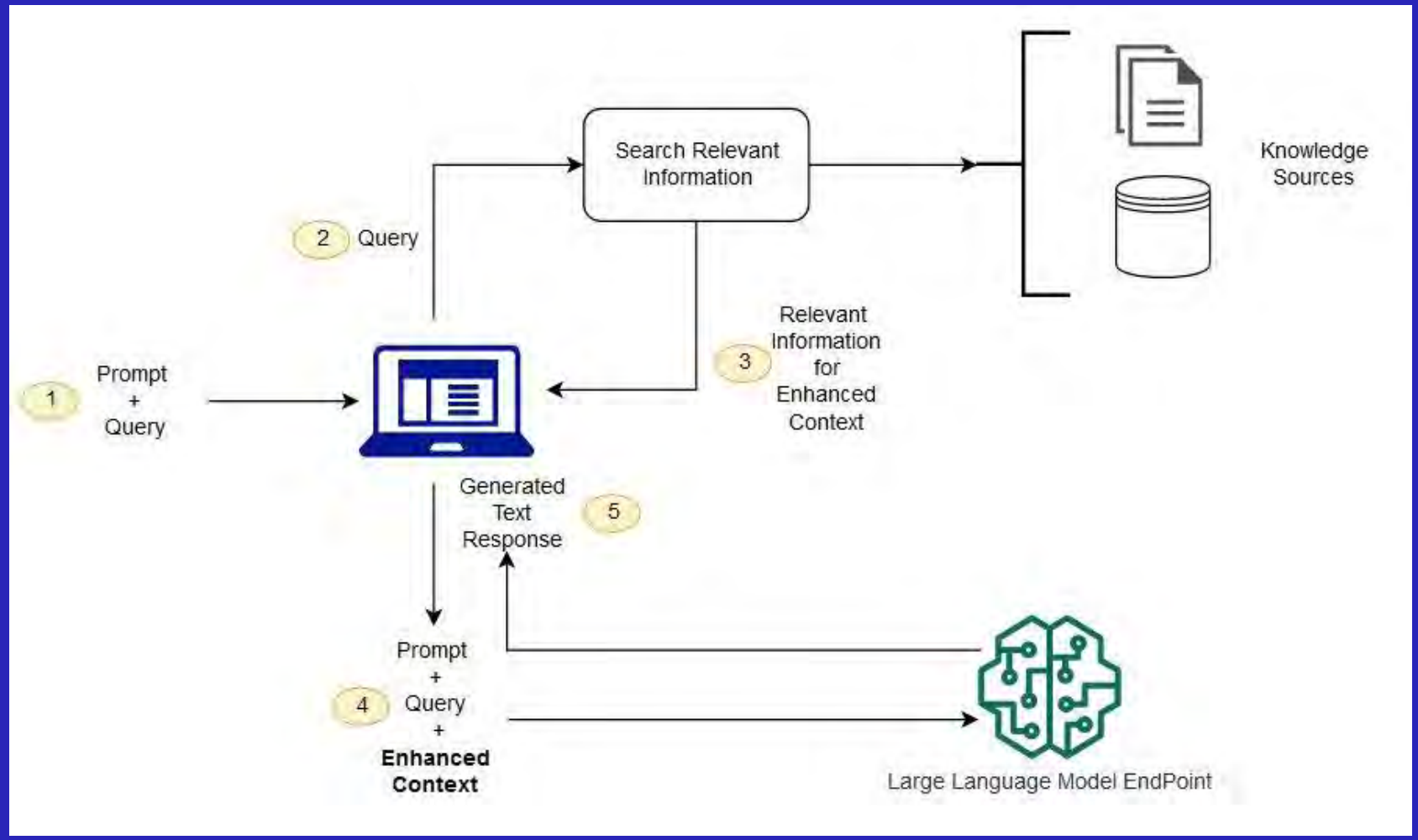
Usage Risks

- Intellectual Property
- Hallucinations
- Copyright

Addressing Model Challenges: Finding Effective Solutions

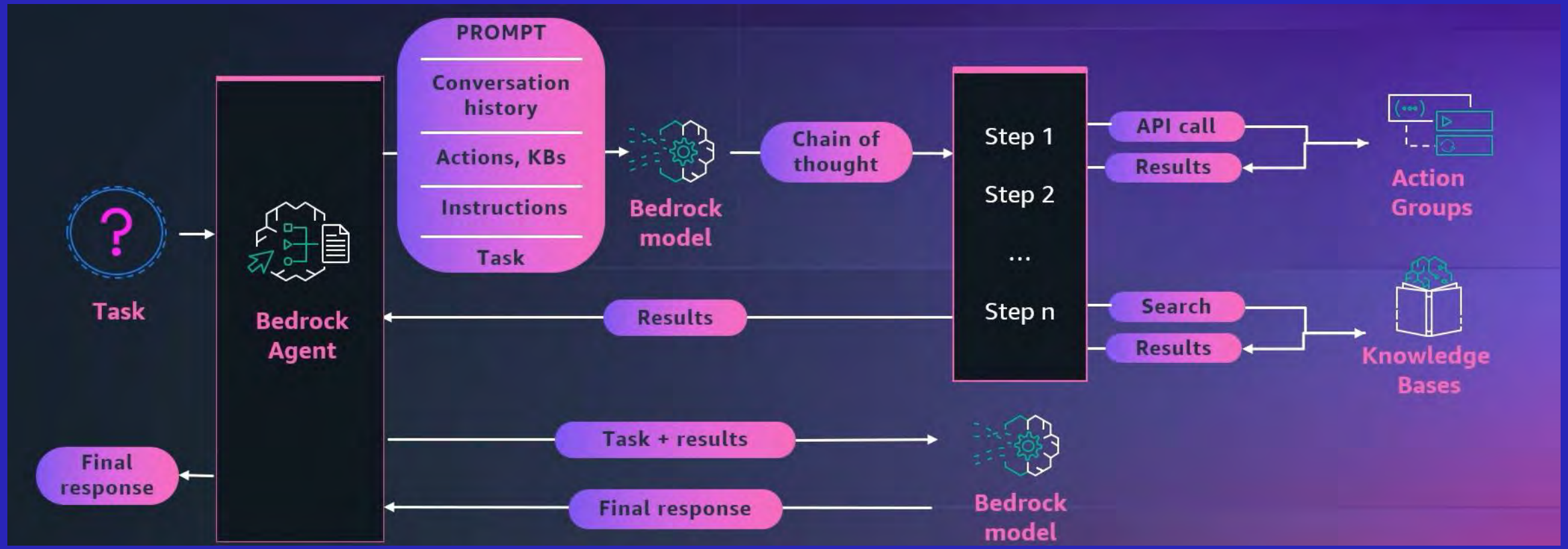
- RAG / Knowledge bases
- LLM Agents
- Prompt Engineering

Retrieval-Augmented Generation (RAG) / Knowledge bases



aws.amazon.com

LLM Agents



aws.amazon.com

Prompt Engineering Best practices

- **Clear Objectives**: Define clear and specific objectives for each prompt to guide model generation effectively.
- **Relevant Context**: Provide relevant context to the model to ensure accurate and meaningful responses.
- **Continuous Evaluation**: Regularly evaluate prompt effectiveness and adjust as needed to improve model performance.
- **Iterative Refinement**: Continuously refine prompts based on model outputs and user feedback for ongoing improvement.
- **Ethical Considerations**: Consider ethical implications when crafting prompts to promote responsible AI usage.

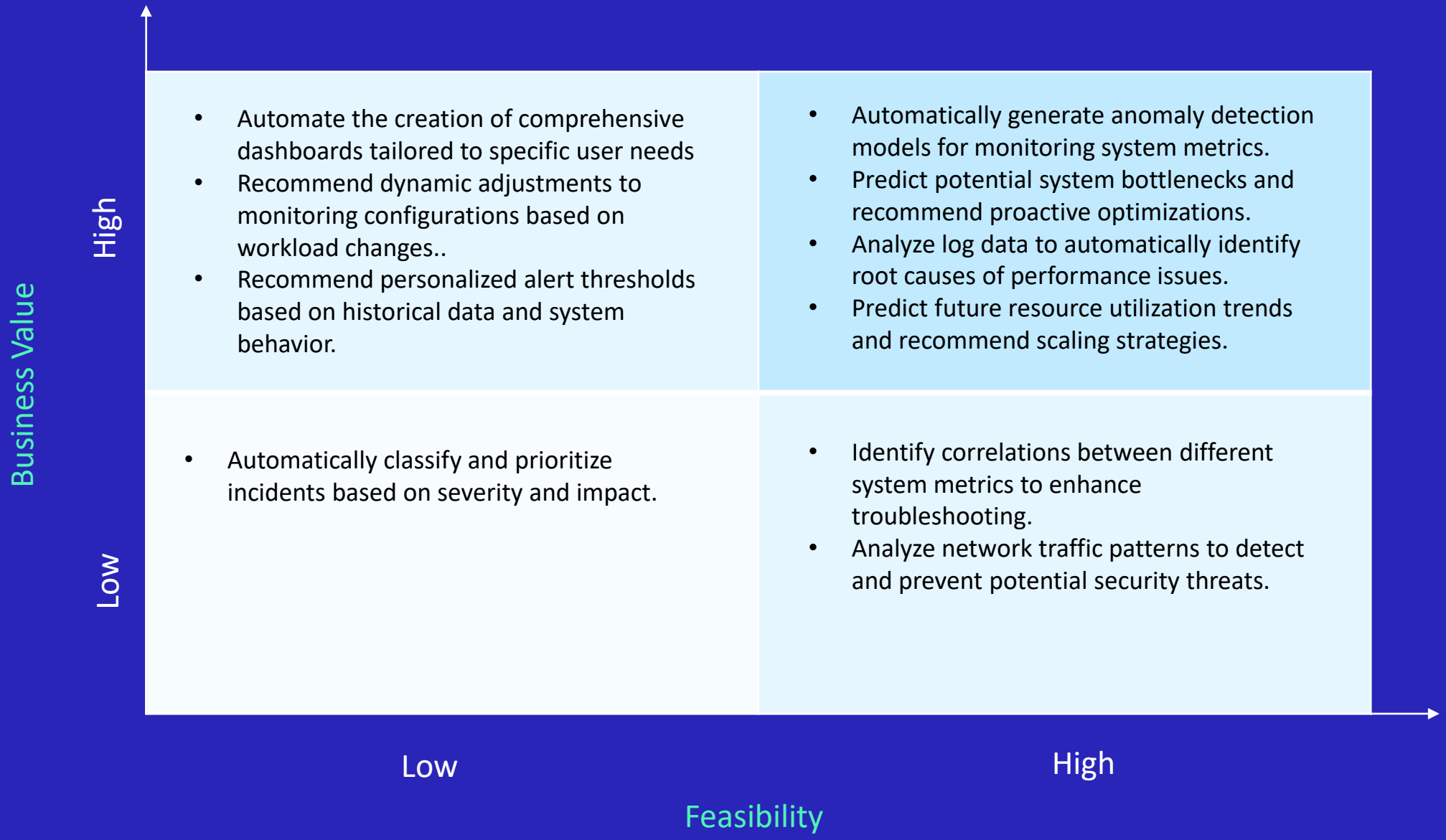
Prompt Engineering Properties

Properties	Details
Temperature	• Controls randomness in model output. Higher temperatures yield more diverse responses; lower temperatures, more focused.
Top-p Sampling	• Controls output diversity by considering only most probable tokens.
Top-k Sampling	• Considers only k most probable tokens for generating next token.
Max Token Length	• Sets maximum length of generated text.
Stop Tokens	• Signals model to stop generating text when encountered.
Repetition Penalty	• Penalizes model for repeating text, encouraging diversity.
Presence Penalty	• Penalizes model for generating already generated tokens.
Batch Size	• Determines number of input sequences processed simultaneously.
Inference Latency	• Time taken for model to generate output given input.
Model Accuracy & Metrics	• Task-specific metrics like accuracy, perplexity, or BLEU score.

SRE 2.0

- GenAI in Observability
- GenAI in SLI, SLO, and Error Budgets
- GenAI in System Architecture and Recovery Objectives
- GenAI in Release & Incident Engineering
- GenAI in Automation
- GenAI in Resilience Engineering
- GenAI in Blameless Postmortems

GenAI in Observability



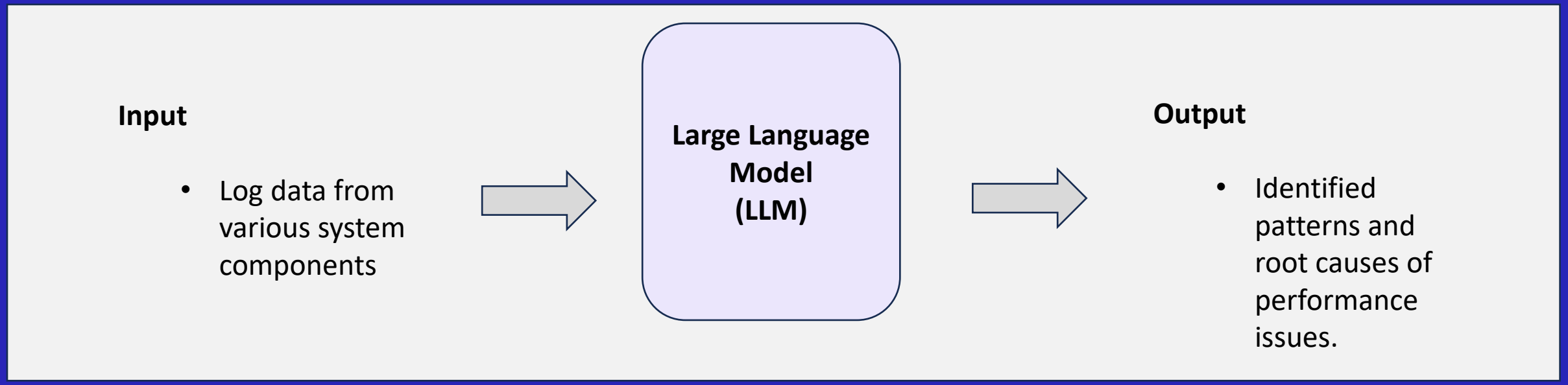
- Automate the creation of comprehensive dashboards tailored to specific user needs
- Recommend dynamic adjustments to monitoring configurations based on workload changes..
- Recommend personalized alert thresholds based on historical data and system behavior.

- Automatically generate anomaly detection models for monitoring system metrics.
- Predict potential system bottlenecks and recommend proactive optimizations.
- Analyze log data to automatically identify root causes of performance issues.
- Predict future resource utilization trends and recommend scaling strategies.

- Automatically classify and prioritize incidents based on severity and impact.

- Identify correlations between different system metrics to enhance troubleshooting.
- Analyze network traffic patterns to detect and prevent potential security threats.

Use Case - Analyze log data to automatically identify root causes of performance issues.



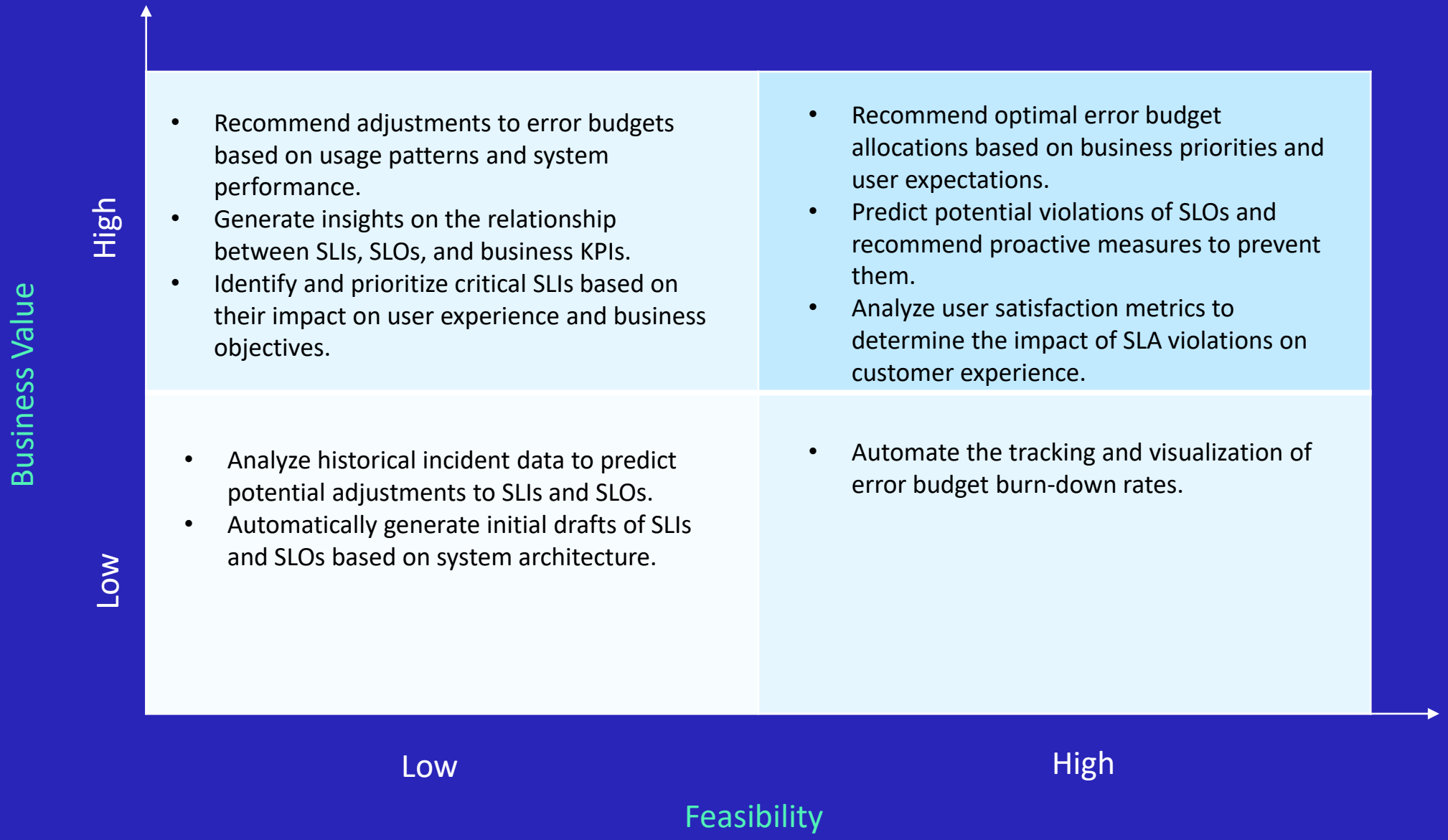
Feedback Loop:

- Human validation of results.
- Feedback from incident resolution.
- Continuous learning.

Ways to improve output:

- Enhancing algorithms.
- Providing domain-specific data.
- Integrating feedback effectively.
- Promoting collaborative human-AI interaction.

GenAI in SLI, SLO, and Error Budgets



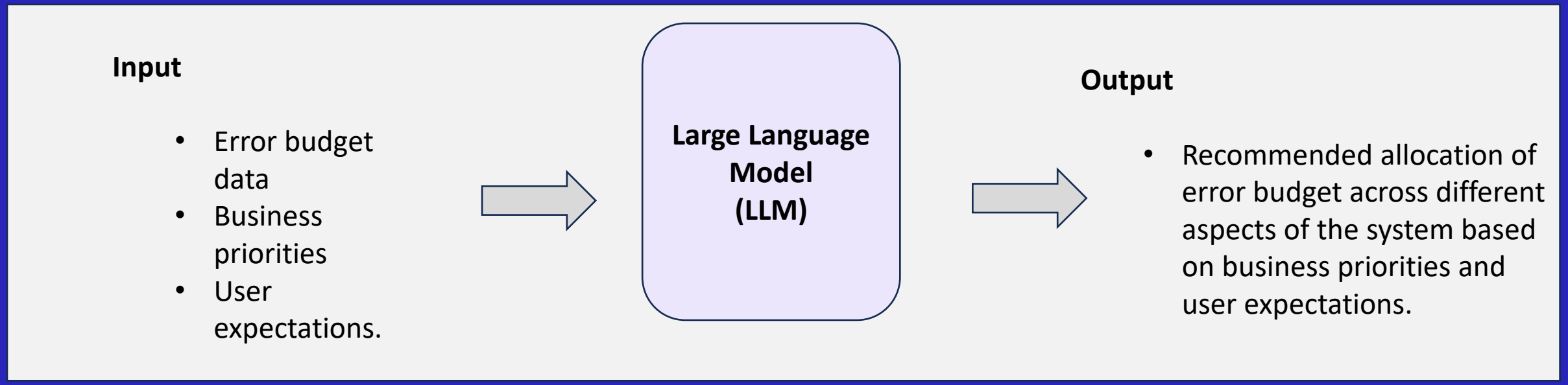
- Recommend adjustments to error budgets based on usage patterns and system performance.
- Generate insights on the relationship between SLIs, SLOs, and business KPIs.
- Identify and prioritize critical SLIs based on their impact on user experience and business objectives.

- Recommend optimal error budget allocations based on business priorities and user expectations.
- Predict potential violations of SLOs and recommend proactive measures to prevent them.
- Analyze user satisfaction metrics to determine the impact of SLA violations on customer experience.

- Analyze historical incident data to predict potential adjustments to SLIs and SLOs.
- Automatically generate initial drafts of SLIs and SLOs based on system architecture.

- Automate the tracking and visualization of error budget burn-down rates.

Use Case - Recommend optimal error budget allocations based on business priorities and user expectations.



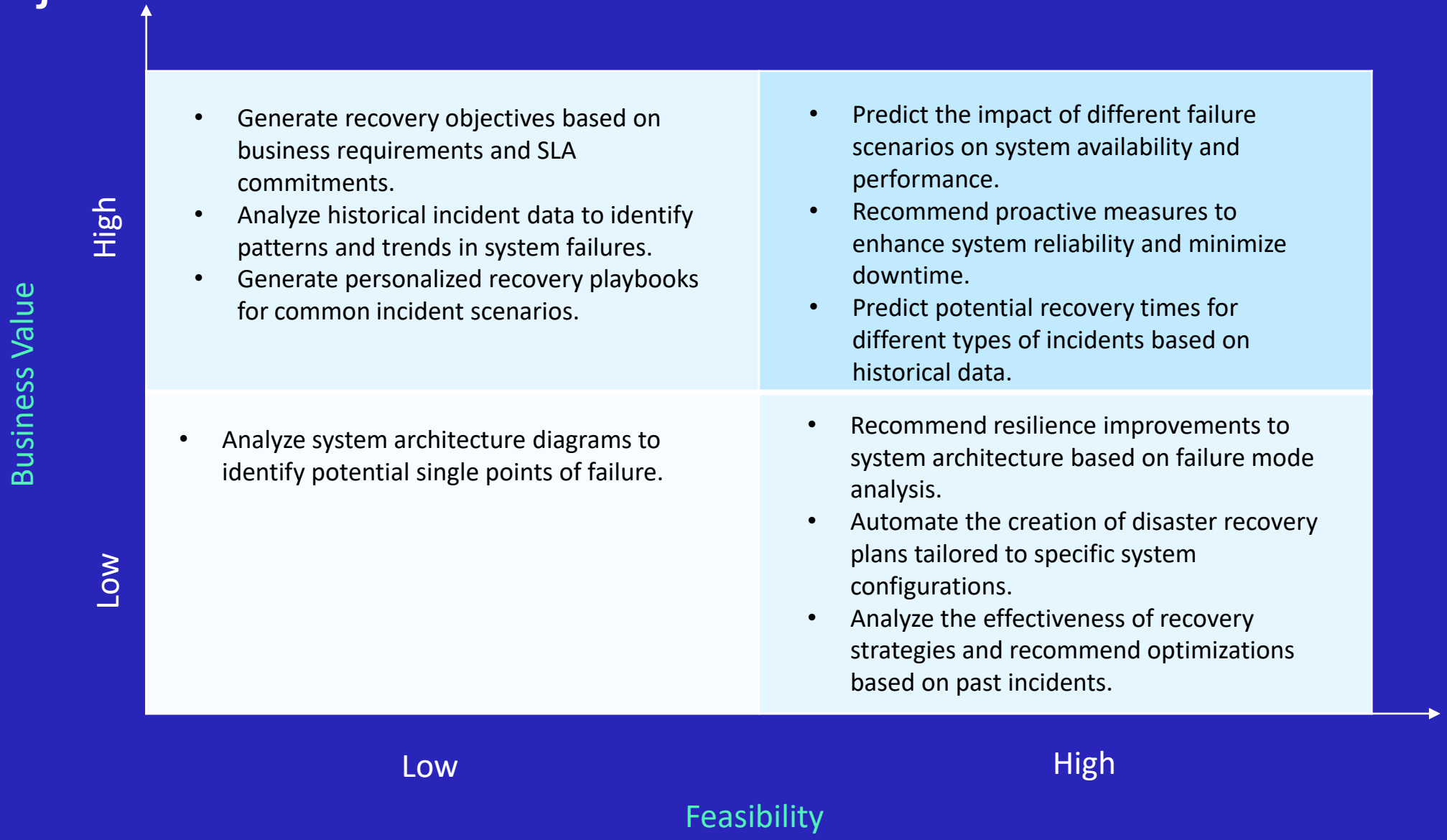
Feedback Loop:

- Evaluation of recommendations by stakeholders.
- Adjustments based on feedback and changing priorities.
- Continuous learning.

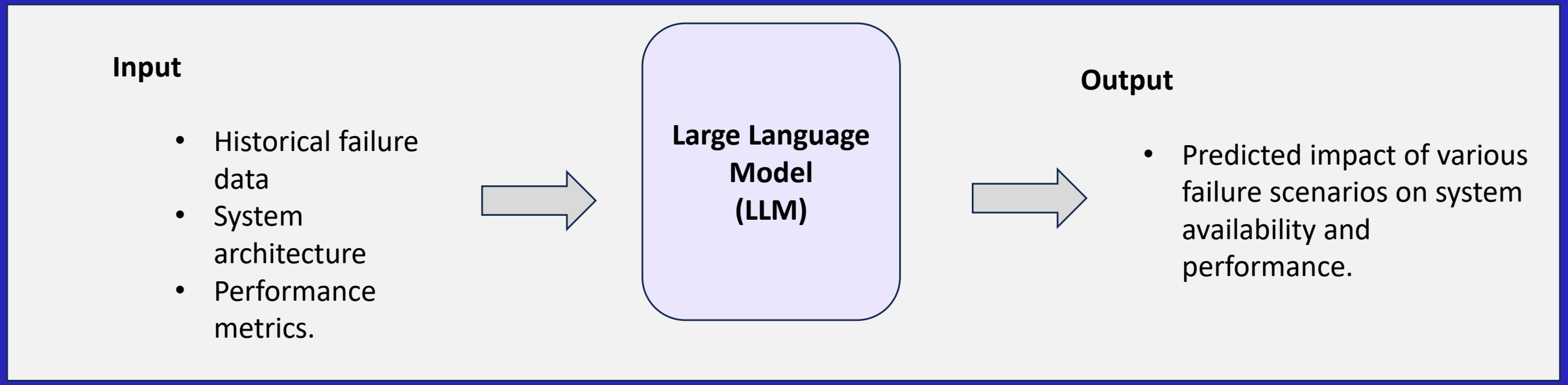
Ways to improve output:

- Incorporating stakeholder feedback into the recommendation process.
- Updating models based on changing business priorities.
- Continuous refinement based on user expectations.

GenAI in System Architecture and Recovery Objectives



Use Case - Predict the impact of different failure scenarios on system availability and performance.



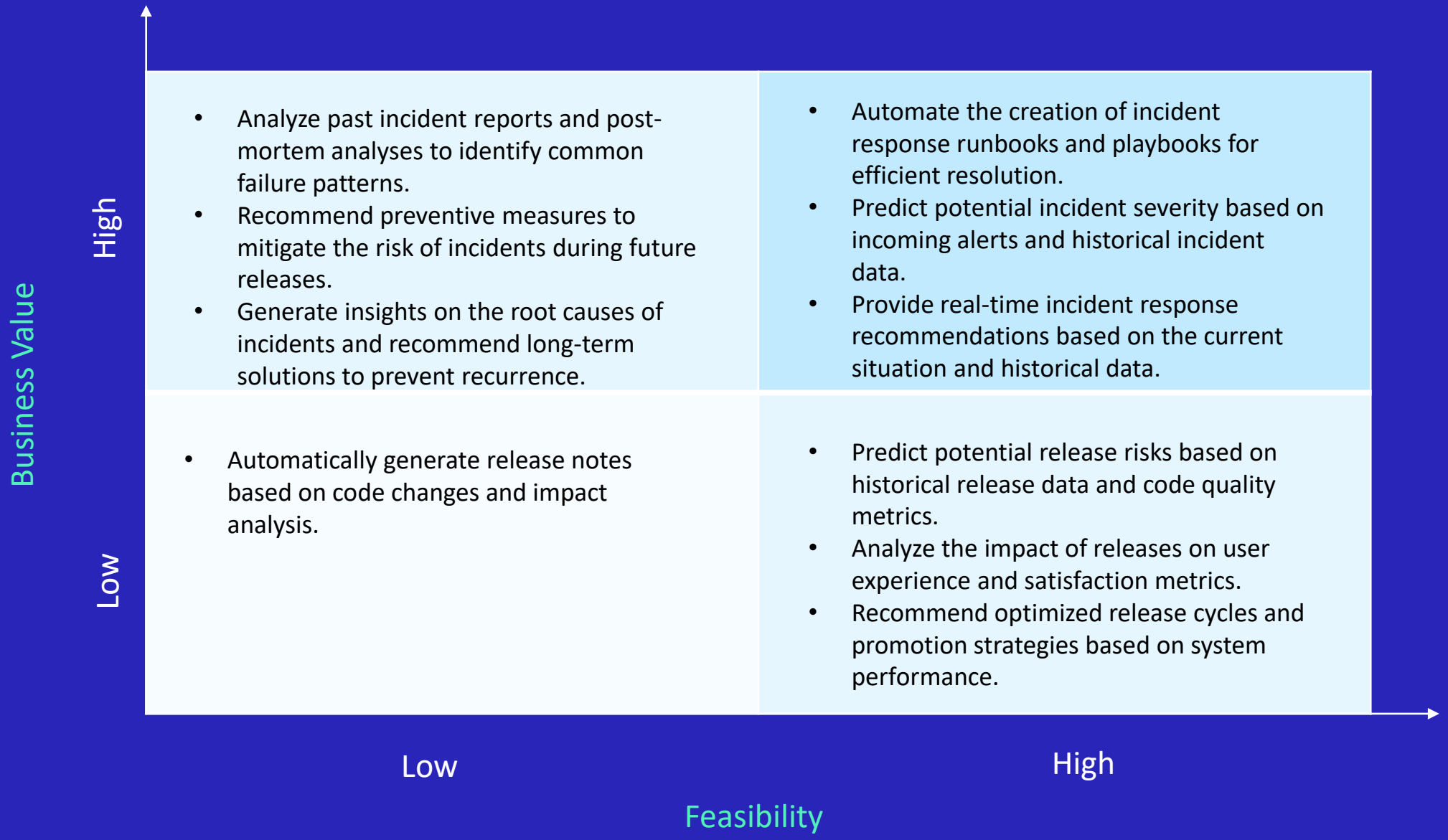
Feedback Loop:

- Validation of predictions against real-world incidents.
- Updating models based on observed impacts and accuracy of predictions.
- Continuous learning from new data.

Ways to improve output:

- Incorporating feedback from incident resolution into prediction models.
- Enhancing models with additional factors influencing system availability and performance.
- Continuous refinement based on observed impacts and feedback

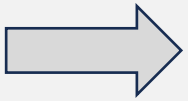
GenAI in Release & Incident Engineering



• Use Case - Provide real-time incident response recommendations based on the current situation and historical data.

Input

- Current incident data
- Historical incident data
- System status and performance metrics.



Large Language Model (LLM)

Output

- Real-time recommendations for incident response actions based on current and historical data analysis

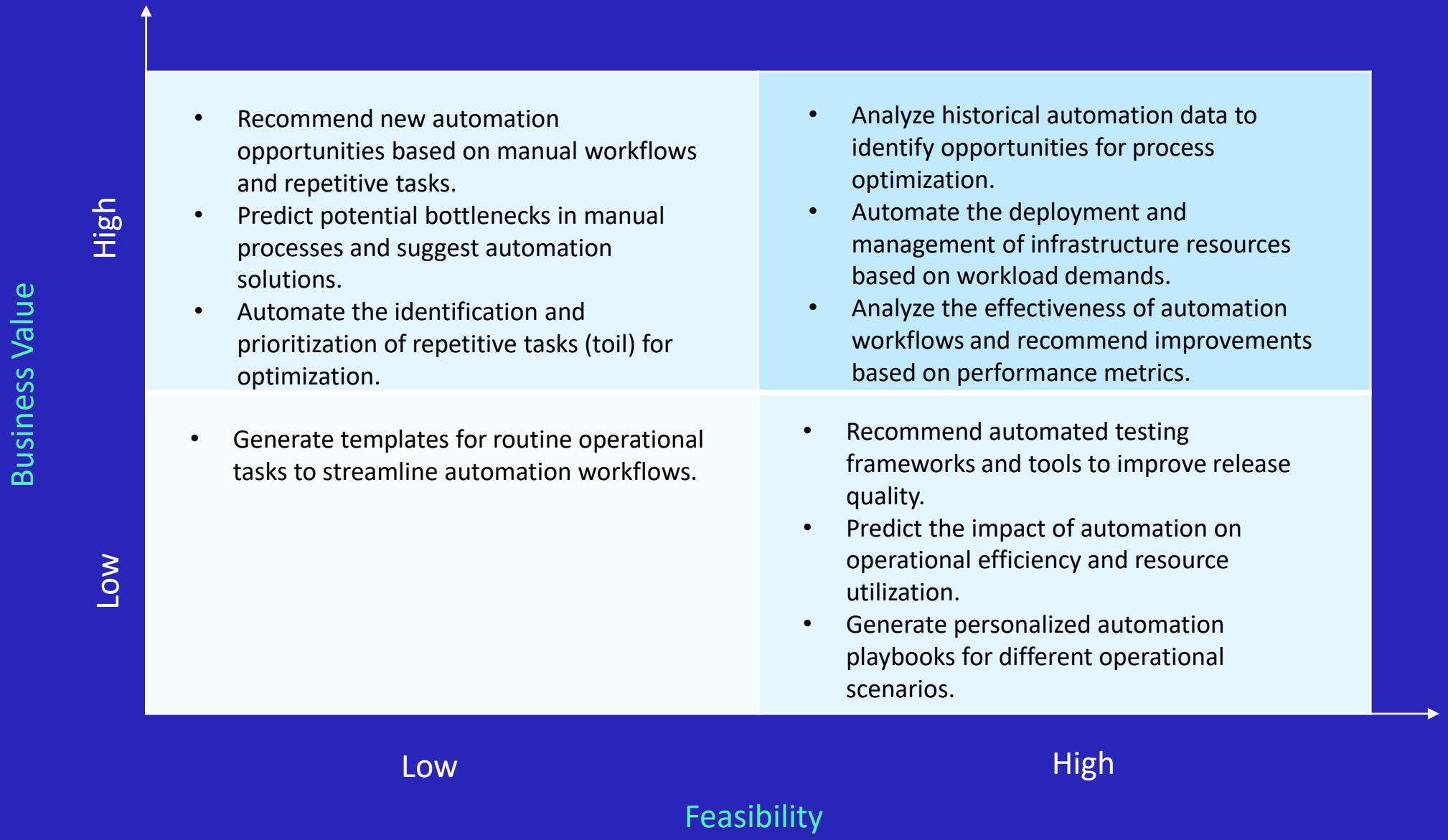
Feedback Loop:

- Evaluation of recommendations' effectiveness in resolving incidents.
- Incorporation of feedback into future recommendations.
- Continuous learning from incident outcomes.

Ways to improve output:

- Integration of real-time feedback from incident resolution into recommendation models.
- Enhancement of models with additional contextual data for more accurate recommendations.
- Continuous refinement based on observed effectiveness and feedback.

GenAI in Automation



- Recommend new automation opportunities based on manual workflows and repetitive tasks.
- Predict potential bottlenecks in manual processes and suggest automation solutions.
- Automate the identification and prioritization of repetitive tasks (toil) for optimization.

- Analyze historical automation data to identify opportunities for process optimization.
- Automate the deployment and management of infrastructure resources based on workload demands.
- Analyze the effectiveness of automation workflows and recommend improvements based on performance metrics.

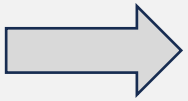
- Generate templates for routine operational tasks to streamline automation workflows.

- Recommend automated testing frameworks and tools to improve release quality.
- Predict the impact of automation on operational efficiency and resource utilization.
- Generate personalized automation playbooks for different operational scenarios.

- Use Case - Analyze the effectiveness of automation workflows and recommend improvements based on performance metrics.

Input

- Automation workflow data.
- Performance metrics.
- Historical data.



Large Language Model (LLM)

Output

- Analysis of automation workflow effectiveness and recommendations for improvements.

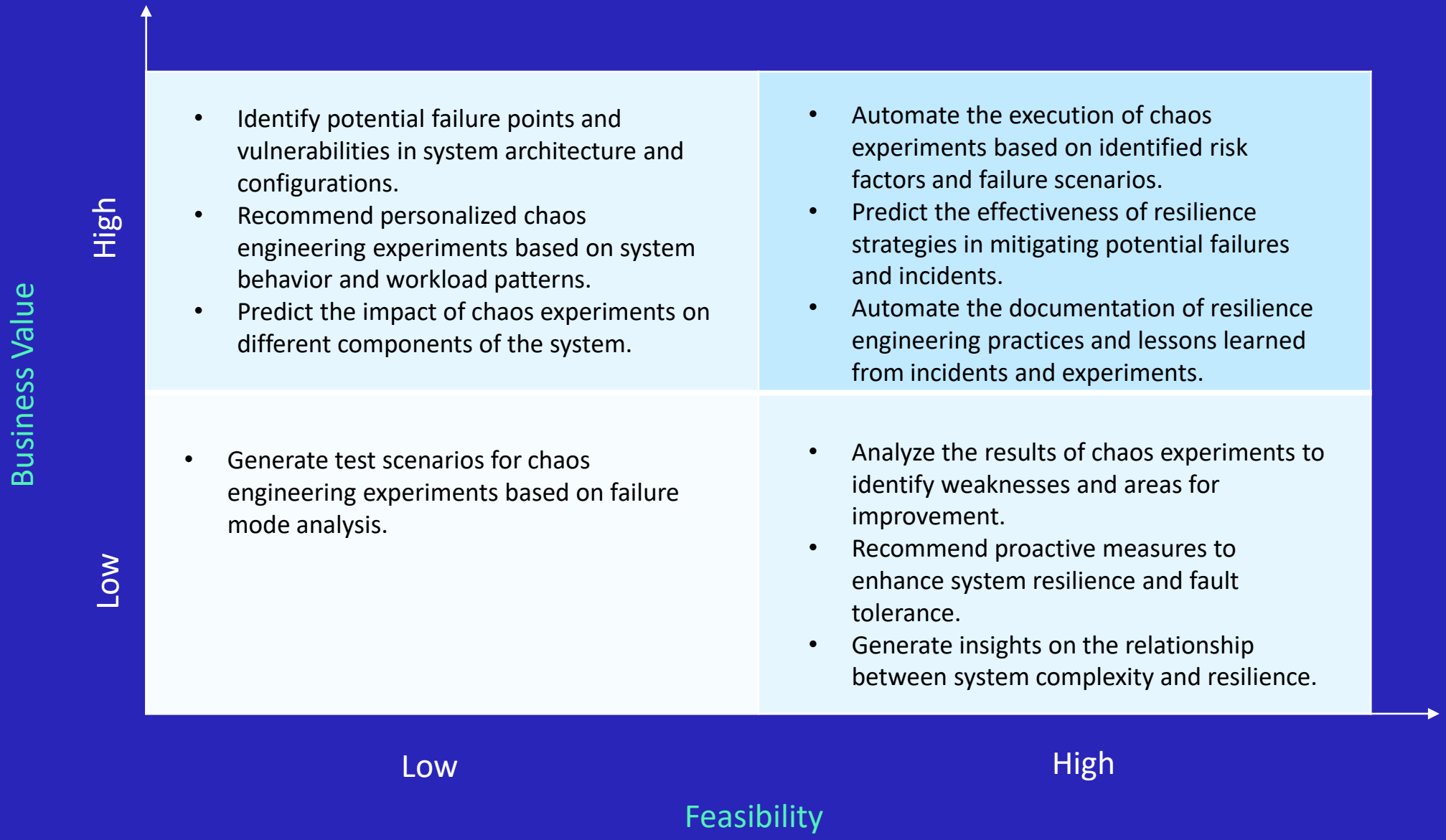
Feedback Loop:

- Evaluation of recommendations' effectiveness in resolving incidents.
- Incorporation of feedback into future recommendations.
- Continuous learning from incident outcomes.

Ways to improve output:

- Evaluation of recommended improvements' impact on workflow performance.
- Incorporation of feedback into future analysis and recommendations.
- Continuous learning from performance outcomes.

GenAI in GenAI in Resilience Engineering



- Identify potential failure points and vulnerabilities in system architecture and configurations.
- Recommend personalized chaos engineering experiments based on system behavior and workload patterns.
- Predict the impact of chaos experiments on different components of the system.

- Automate the execution of chaos experiments based on identified risk factors and failure scenarios.
- Predict the effectiveness of resilience strategies in mitigating potential failures and incidents.
- Automate the documentation of resilience engineering practices and lessons learned from incidents and experiments.

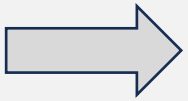
- Generate test scenarios for chaos engineering experiments based on failure mode analysis.

- Analyze the results of chaos experiments to identify weaknesses and areas for improvement.
- Recommend proactive measures to enhance system resilience and fault tolerance.
- Generate insights on the relationship between system complexity and resilience.

- Use Case - Automate the execution of chaos experiments based on identified risk factors and failure scenarios.

Input

- Identified risk factors
- Failure scenarios
- System architecture
- Chaos experiment parameters



Large Language Model (LLM)

Output

- Automated execution of chaos experiments targeting identified risk factors and failure scenarios.

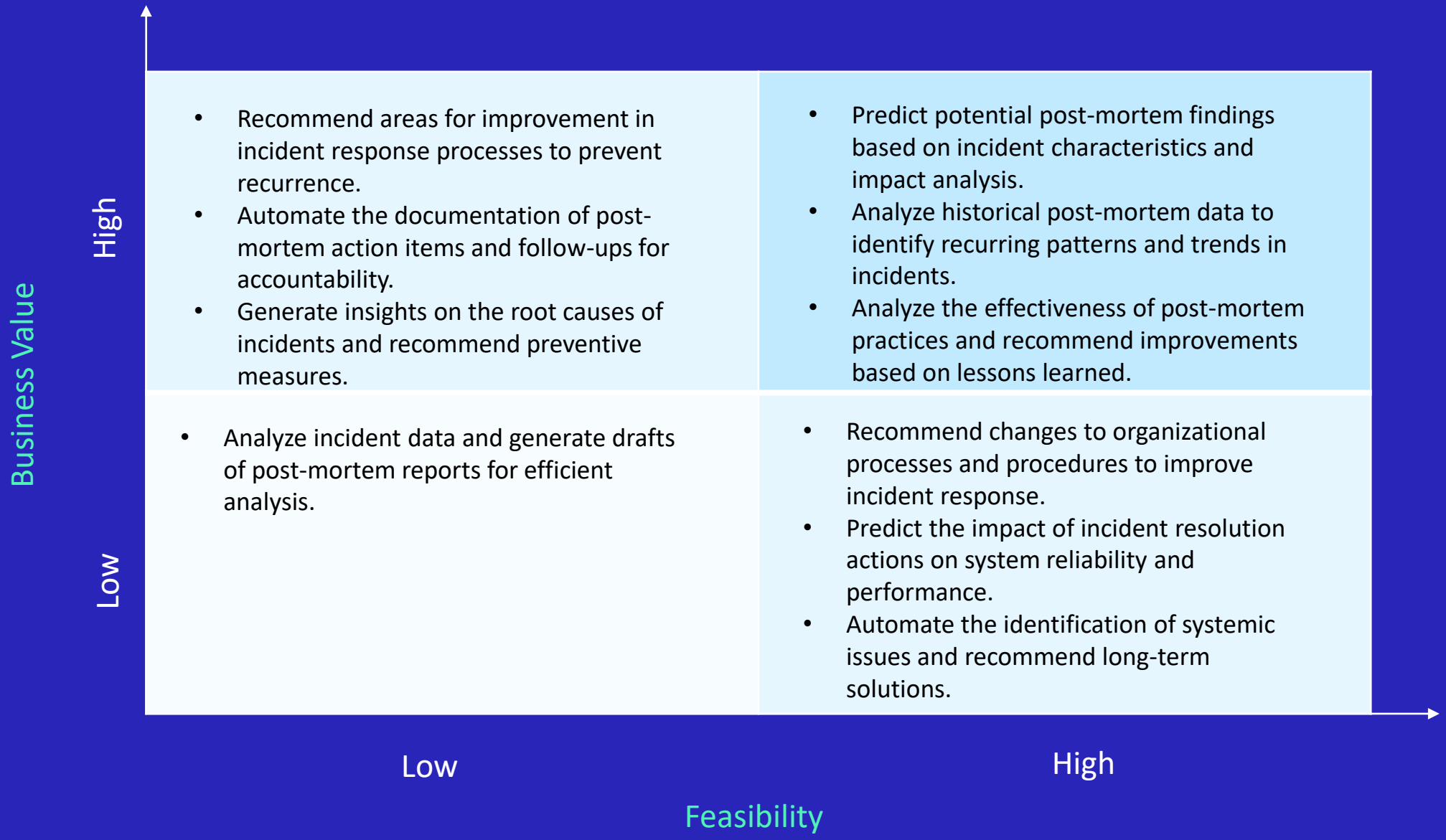
Feedback Loop:

- Evaluation of the impact of chaos experiments on system behavior and stability.
- Incorporation of feedback into future chaos experiment automation.
- Continuous learning from experiment outcomes.

Ways to improve output:

- Integration of real-time feedback from chaos experiments into automation algorithms.
- Enhancement of automation models with additional contextual data for more accurate experiment selection.
- Continuous refinement based on observed outcomes and feedback.

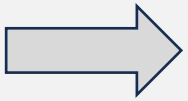
GenAI in GenAI in Blameless Postmortems



• Use Case - Analyze historical post-mortem data to identify recurring patterns and trends in incidents.

Input

- Historical post-mortem data
- Incident reports
- Root cause analyses



Large Language Model (LLM)

Output

- Identification of recurring patterns and trends in incident occurrences and root causes.

Feedback Loop:

- Validation of identified patterns against real-world incidents.
- Incorporation of feedback into future analysis and pattern recognition.
- Continuous learning from incident data.

Ways to improve output:

- Integration of real-time feedback from incident resolution into pattern recognition algorithms.
- Enhancement of analysis models with additional contextual data for more accurate trend identification.
- Continuous refinement based on observed patterns and feedback.

Measure Progress with Business Outcomes

- Improved Alignment with SLOs and Error Budgets
- Increased Change Frequency
- Reduced Change Failure Rate
- Reduced Lead Time for Change
- Reduced Mean Time to Detect (MTTD)
- Reduced Mean Time to Repair (MTTR)
- Increased Mean Time Between Failures (MTBF)

Best practices

- **Clear Objectives:** Define specific goals for each use case to guide the generative AI process effectively.
- **Relevant Context**: Provide accurate and relevant data and context to ensure the generated outputs align with real-world scenarios.
- **Continuous Evaluation**: Regularly assess the quality and effectiveness of generated outputs and refine the models based on feedback.
- **Ethical Considerations**: Consider ethical implications in the use of generative AI, ensuring outputs are fair, unbiased, and respect privacy.
- **Collaborative Approach**: Involve domain experts and end-users in the generative AI process to incorporate diverse perspectives and improve outcomes.
- **Documentation:** Document the generative AI process, including data sources, model architectures, and decision-making criteria, for transparency and reproducibility.
- **Iterative Improvement**: Continuously iterate on the generative AI models based on insights gained from real-world implementation and user feedback to enhance performance and reliability.

Pitfalls to Avoid

- **Ignoring Ethical Considerations**: Neglecting to address potential biases or ethical concerns in generated outputs can lead to unintended consequences.
- **Lack of Validation**: Failing to validate generated outputs against real-world data or expert judgment may result in inaccurate or irrelevant recommendations.
- **Disregarding User Feedback**: Ignoring feedback from end-users or domain experts can lead to mismatches between generated outputs and actual user needs.
- Static Models: Avoid treating generative AI models as static solutions; instead, regularly update and refine them to adapt to evolving requirements and environments.

Thank you.