nagarro

Webinar

Intelligent strategies for tomorrow’s testing

Agenda

Lightning Talks

1. **Testing of AI-based applications**
   Vimmi Walia

2. **Self-healing pipelines**
   Manisha Mittal

3. **Accessibility Testing**
   Anamika Mukhopadhyay

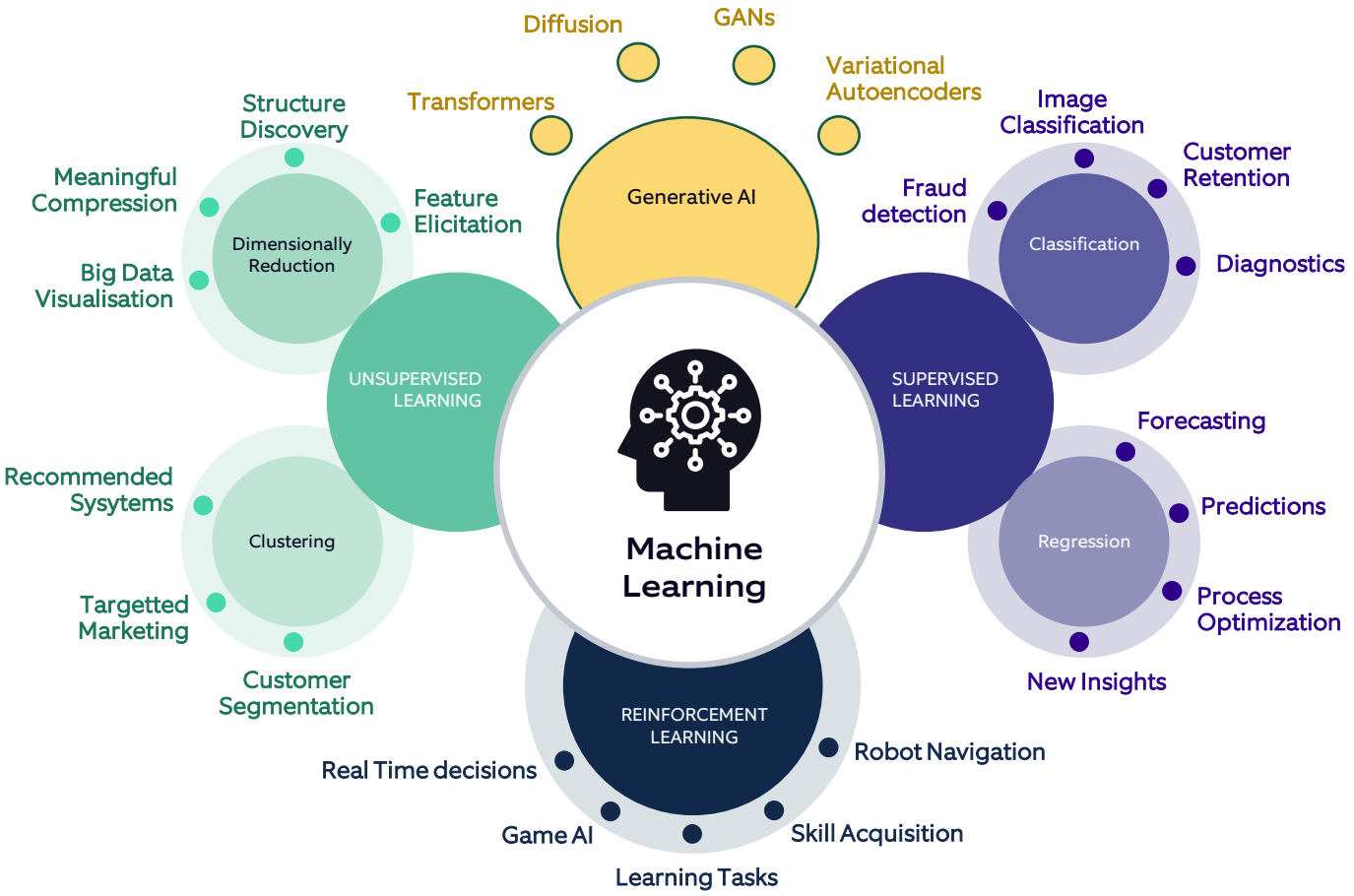
nagarro

Intelligent strategies for tomorrow‘s testing

Testing of AI-based applications

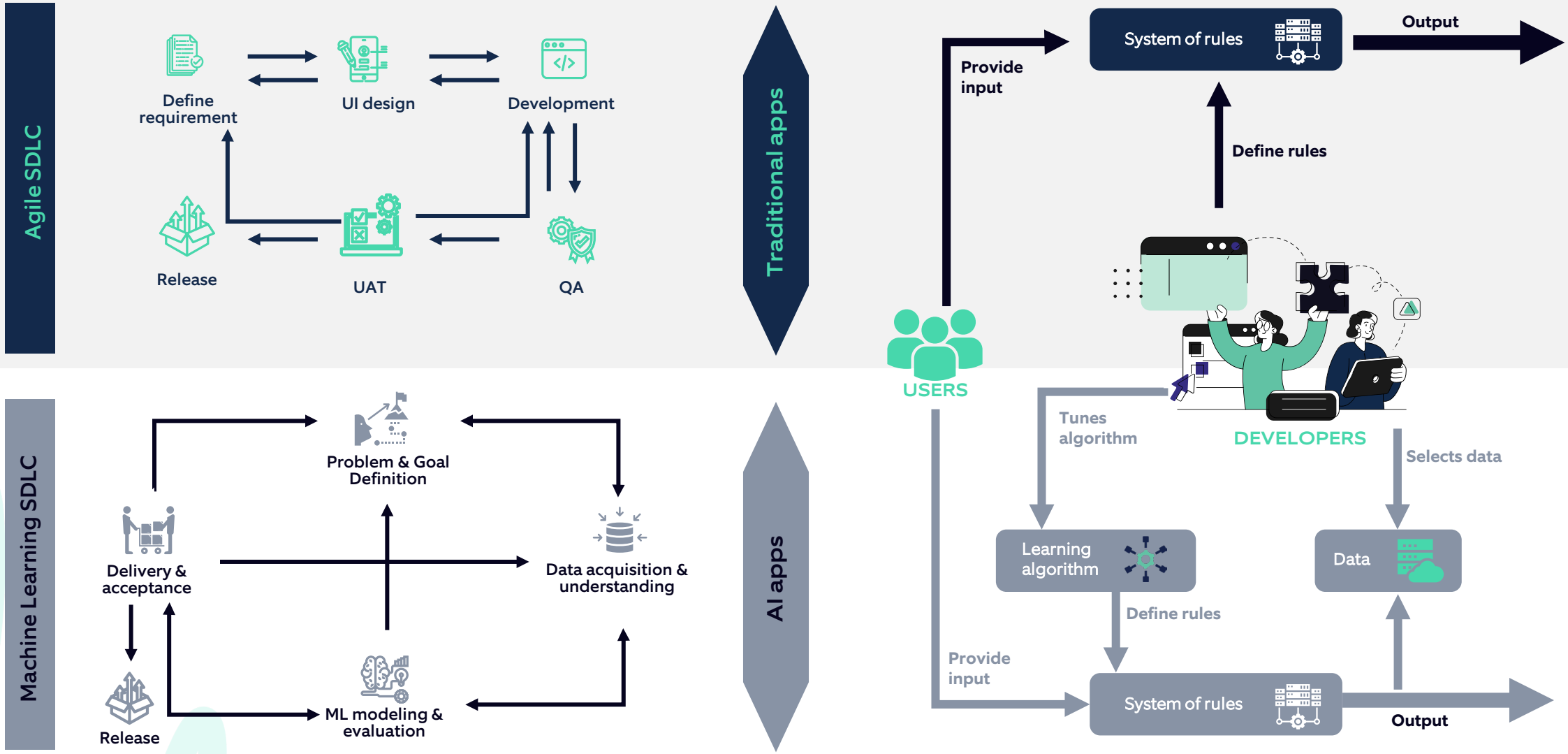
Vimmi Walia

An insight into the AI world

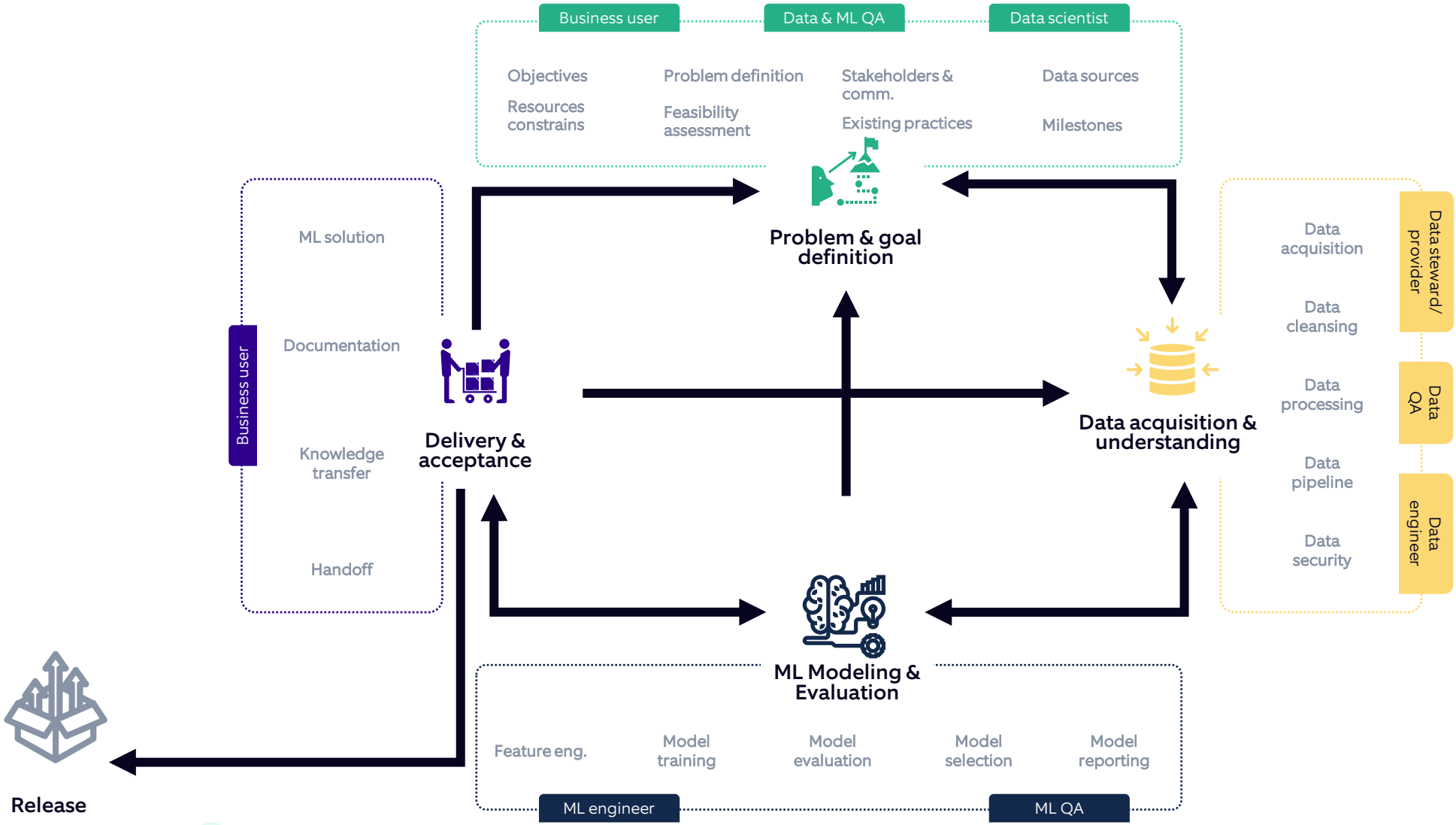


Compute in AI has been doubling every 3.4 months since 2012.	By 2030, 70% of companies will adopt some AI.	AI could potentially add up to 16%, approximately $13 trillion, to the global economy by 2030.	AI front-runners will double their cash-flow by 2030.	AI nonadopters and slow adopters lose -20% of their cashflow by 2030.

Traditional apps vs AI apps



AI SDLC – Zoom-in



AI is not black box

01
Data collection

02
Data processing

Supposedly assumed backbox

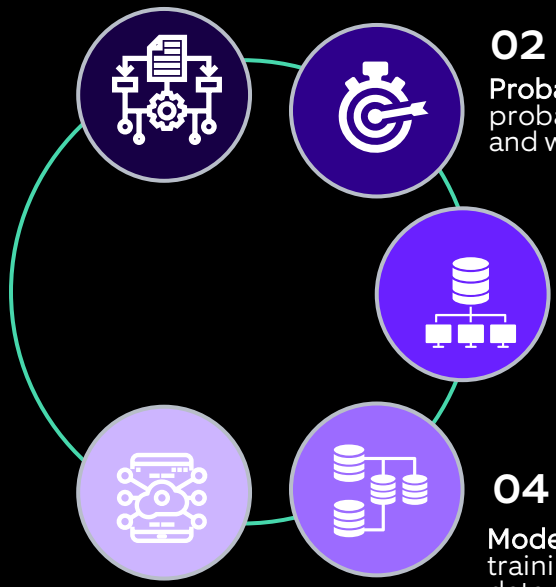
03
AI : EDA, Feature Engineering, Model evaluation, Model selection

04
Data output

05
Data visualization

01

Algorithms : AI is a collection of complex mathematical functions/algorithms like Linear regression, Decision trees, Naive Bayes.



02

Probability: AI applications are based on probabilistic logic, hence the results are not black and white. We talk in term degree of accuracy.

03

Data: And it needs lots and lots of data to learn the required pattern using the chosen algorithm and model the required behaviour.

04

Model Training: Model is first trained with the training data set. To validate the model test data set is executed on the model.

05

Model drift: Many AI models evolve with time with the new data sets coming into the application.

Key takeaways

Customized test approach for every use case

The appropriate evaluation metrics can vary depending on the application context, requiring domain-specific knowledge to choose the right metrics.

Lack of determinism

Exhibit non-deterministic behavior. They may produce different results for the same input under different conditions or over time as they learn and adapt.

Data quality

Ensuring the quality, relevance, and representativeness of the data is critical.

Bias

Source Data may be biased, and the data used to train an AI system is not representative of the reality it's meant to model.

Model and data drift

The data that the model was trained on becomes outdated or no longer represents the current conditions.

Explainability

Many AI models, particularly deep learning models, are often referred to as "black boxes" due to their opaque decision-making process.

nagarro

Intelligent strategies for tomorrow‘s testing

Automated anomalies detection with self-healing pipelines

Manisha Mittal

Problem statement

Are you facing these challenges in CI/CD & scared of losses which can occur due to OUTAGES ?

- Manual intervention for error detection and resolution
- Manual inspection of data & logs is unmanageable & time consuming
- Limited scalability with complex systems
- Increased downtime and service disruptions
- Inefficient resource utilization
- Risk of human error and operational inefficiencies

Self-healing pipeline model

Self-healing model aims at an innovative approach for addressing the challenges faced by traditional deployment techniques.

These pipelines are designed to autonomously identify and resolve issues without requiring manual intervention.

By leveraging automation and intelligent algorithms, self-healing model can significantly improve the efficiency and reliability of the software development process

- Transitioning from reactive to proactive approach.
- Automating error detection and resolution.
- Embracing AI/ML for intelligent monitoring.
- Enhancing system resilience and reliability.

Key features of the DevOps self-healing model

01

Anomaly detection

Monitor ->Detect ->Trigger

Structured pipeline logs are used as input to AI-based anomaly detection algorithms to identify unusual or unexpected patterns in pipeline executions like sudden failures. For example, AI model can detect the failure cause of a pipeline execution like network issues, test timeout errors and many more.

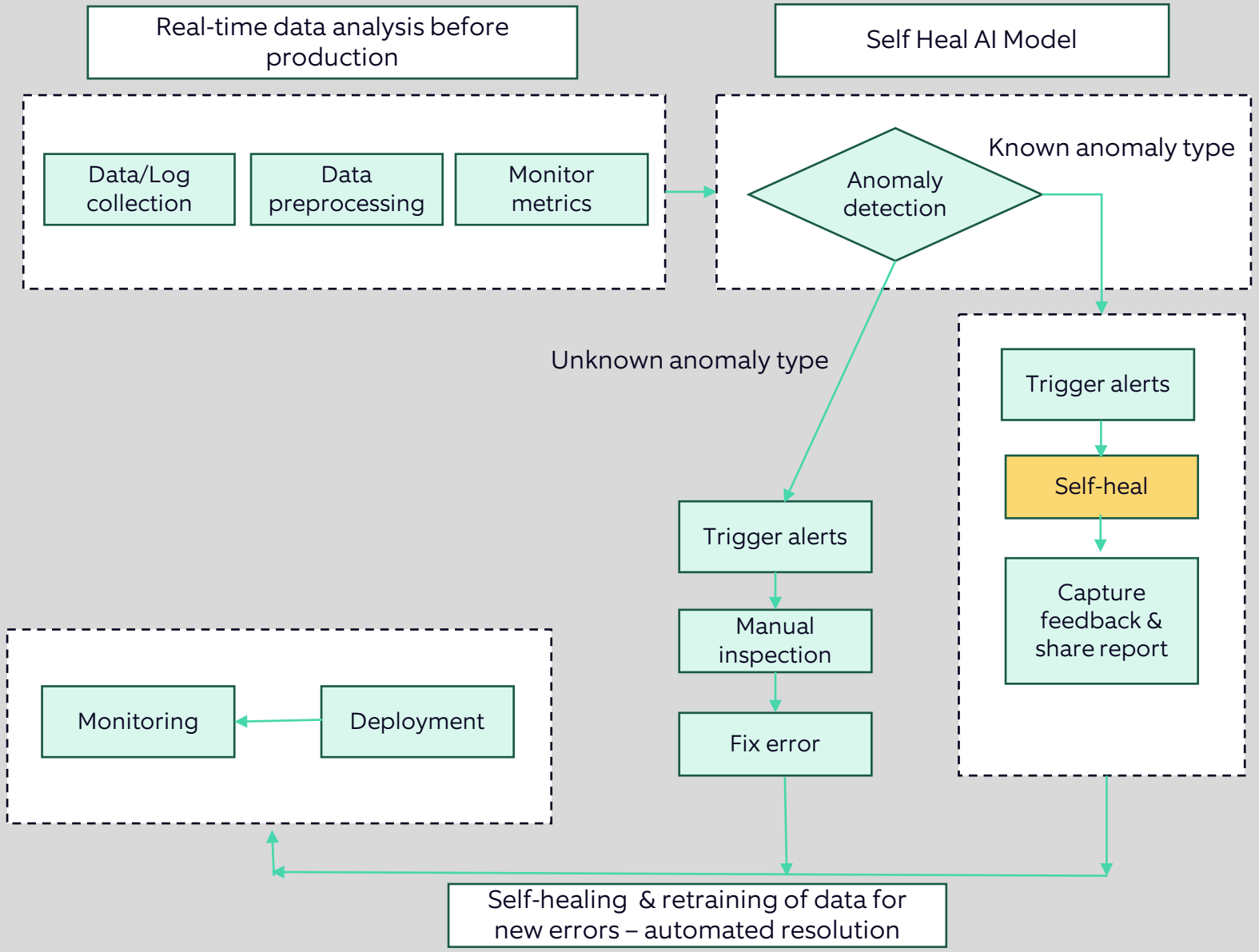
02

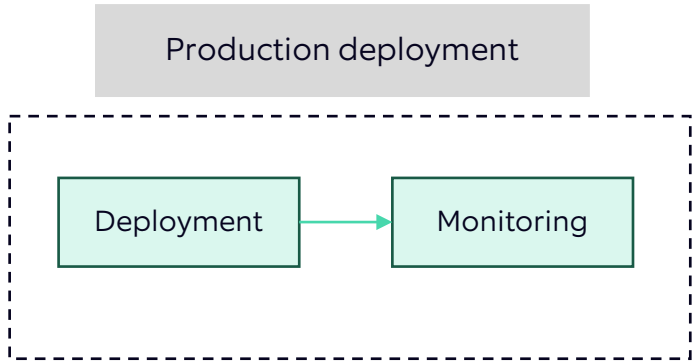
Anomaly remediation

Analysis ->Remediation ->Validation ->Feedback loop

AI algorithms leverages historical data to automate remediation actions in response to detected anomalies or performance degradation. For example, restarting failed processes, reconfiguring system parameters, triggering alternative workflows and fallback mechanism & rollbacks.

How it works





The data and logs generated during **development** and **staging** are used to identify anomalies before releasing into production, while those in the **production** environment can be used to prevent system outages.

Self-healing pipeline scenarios

Network or connectivity issues
Pipeline execution fail if there are network issues or connectivity problems between CI/CD tools and external resources, such as source code repositories, artifact repositories, or external services.

Insufficient permissions
If the CI/CD tools user or service account does not have sufficient permissions to access required resources or perform specific actions within the pipeline, execution may fail due to permission errors.

Resource constraints
If the CI/CD tools master or agent nodes experience resource constraints, such as CPU or memory limitations, pipeline execution fail due to resource exhaustion or timeouts.

Timeouts
If certain stages or steps in the pipeline take longer than the configured timeout threshold, CI/CD tools terminates the pipeline execution, resulting in failure.

Environment configuration issues
Pipeline execution may fail if there are issues with the configuration of the execution environment, such as incorrect environment variables, incompatible software versions, or missing dependencies.

Version control issues
If there are conflicts, merge failures, or other issues with version control repositories (e.g., Git, SVN) that the pipeline relies on for source code, pipeline execution may fail.

External service failures
If the pipeline interacts with external services, APIs, or webhooks, failures or downtime in those services can cause pipeline execution to fail.

Script security restrictions
If CI/CD tools is configured with strict script security settings, pipeline execution fail if the pipeline script violates security policies or tries to execute disallowed operations.

Challenges in achieving the self-healing

Cultural shift

Adopting a self-healing mindset necessitates a cultural shift, trusting automated systems with critical tasks, and embracing proactive problem-solving.

Security concerns

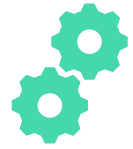
Automated responses raise security concerns, particularly regarding access control and authorization, demanding that self-healing mechanisms prioritize system security.

Algorithm selection

Selecting appropriate algorithms for anomaly detection and decision-making is complex, considering different issues and use cases.

Accurate decision

Ensuring precise anomaly and issue detection is vital for effective self-healing to avoid unnecessary interventions or overlooked problems impacting system reliability.



Monitoring and feedback loops

Establishing robust monitoring and feedback loops is crucial for refining self-healing mechanisms through continuous system performance monitoring and user feedback analysis.

Key takeaways

Autonomous problem solving: Detects and corrects issues without human intervention

Resilience and reliability: Reduces downtime, ensures smooth operations

Cost-efficiency: Saves time, resources, and mitigates risks

Scalability: Adapts to changing demands seamlessly

Continuous improvement: Feedback loops & iterative improvements

nagarro

Intelligent strategies for tomorrow‘s testing

Accessibility Testing

Anamika Mukhopadhyay

What is Accessibility?

- Refers to the design and development of products, services, and environments that can be accessed and used by individuals with disabilities

- It aims to eliminate barriers and provide equal access and opportunities for everyone, regardless of their abilities

No longer a choice, it’s a requirement!

Tomorrow's testing need

An estimated 1.3 billion people experience significant disability. This represents 16% of the world's population, or 1 in 6 of us (WHO).

The accessibility market is expected to grow at a compound annual growth rate (CAGR) of 6.20 % from 2023 to 2028.

In 2020, companies faced over 3,500 website accessibility lawsuits in the United States, according to data from Usable.net.

3.7% of the world's top 1 million website homepages are WCAG 2.1, level 2 compliant (WebAIM million).

Why is Accessibility needed ?

User experience

Legal compliance

Equal access

Social responsibility

Business benefits

Innovation and creativity

The power of the web is in its universality. Access by everyone regardless of disability is an essential aspect.

-Tim Berners-Lee, Director of the W3C and inventor of the web

Some consequences of lack of a11y

2005

NATIONAL FEDERATION OF THE BLIND
Live the life you want.
vs
target

2012

National Association of the Deaf
vs
NETFLIX

2017

Dennis Haynes
vs
HOOTERS

2017

Maria Mendizabal
vs

2018

Himelda Mendez
vs

2019

Guillermo Robles
vs
Domino's

2022

Rafael Cordero
vs
Electronic Arts

Reference:

- https://www.whoisaccessible.com/guidelines/largest-web-accessibility-lawsuits/
- https://topclassactions.com/lawsuit-settlements/lawsuit-news/apple-class-action-lawsuit-says-website-inaccessible-blind/

Laws/Regulations

US
United States

Americans with Disabilities Act (ADA)

GB
United Kingdom

Disability Discrimination Act (DDA)

EU
European Union

European Accessibility Act (EAA)

CA
Canada

Accessible Canada Act (ACA)

AU
Australia

Disability Discrimination Act (DDA)

Testing for success: A comprehensive accessibility strategy

Manual testing

Making use of assistive technology like screen readers, speech recognition software and special adaptive keyboards

Color contrast analyzers to identify color and contrast related issues

HTML/CSS parsers, bookmarklets, PDF accessibility checkers for code review

Keyboard-only accessibility to ensure that each element of the page is accessible through the keyboard

Accessibility testing strategy

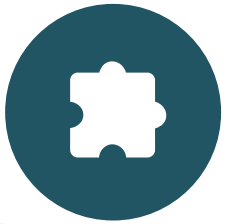
Automated testing

Integrate plug-ins like Axe into the development workflow to expedite testing process

Browser extensions like Axe, Wave for accessibility issues

Command line tools like Axe CLI to perform automated accessibility testing and execute repetitive tasks

Incorporate artificial intelligence to analyze and identifying accessibility issues



Tools and Domains

Automated Testing Tools

Web

- Accessibility Insights
- Wave
- Axe
- Tenon
- pa11y
- Lighthouse

Browser Extensions

- Accessibility Insights
- Wave
- Axe

Mobile

- Accessibility Scanne
- Axe Dev Tools

Manual Testing Tools

Screen Readers

- NVDA
- JAWS
- Talkback
- Voiceover

Colour Contrast

- WCAG Color Contrast

Domains

- BFSI
- Telecom
- Travel and Logistics
- Aviation
- Automotive
- Ecommerce
- Energy and Utilities
- Entertainment
- Life Sciences and Healthcare
- Public Sector
- ISV
- Media and Publishing

By 2025, the number of differently abled people employed will triple, reducing barriers to access

Let's work together to make a positive impact on our customers and our community.

nagarro

Thank you

aqt@nagarro.com