



S
P
I
D
E
R

NAI Software Engineering

Team Introduction & Overview

Enter the Spider-Verse

sky max

Agenda

- 1 Introduction
- 2 The Team
- 3 Our Journey
- 4 Projects
- 5 Agile Methodology
- 6 DevOps Lifecycle
- 7 TDD/BDD Approach
- 8 Conclusion

Our Purpose

Software, Quality Assurance & DevOps



We deliver, test, maintain and optimise high-value applications
through best practises, to support Sky Group's business needs.

SDLC, Industry Standards
& Agile Methodology



Planned Inventory, IPAM, TSA,
Wholesale Ethernet, uSDP



Our Journey

Scaling the Team

How it started

Early days of the team



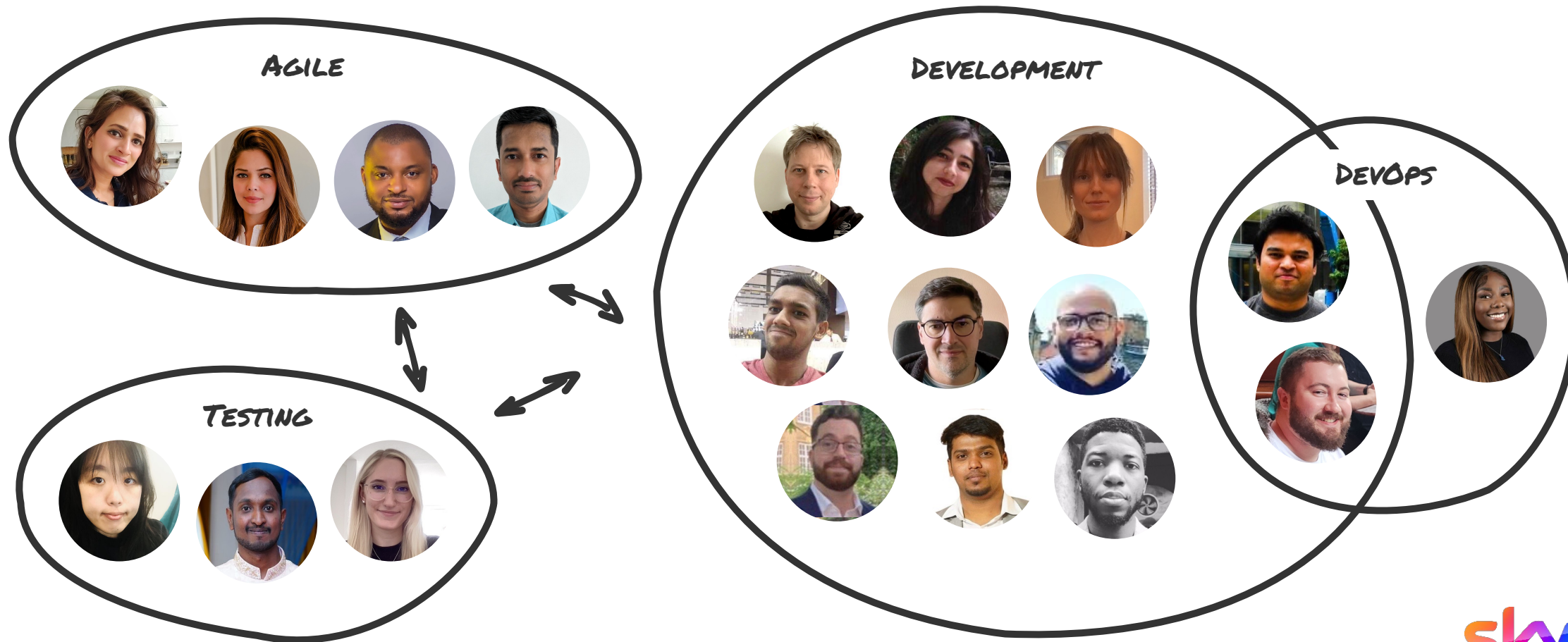
← IBRAHIM

OASIS DEV TEAM



How it's going

Current state of the team



The Team

Who we are



Emanuele Manco

Software Engineering Manager

Millennial from Italy.
Former draughtsman & designer.

Over two decades of experience in the entertainment, fashion, government, finance and telecom industries.

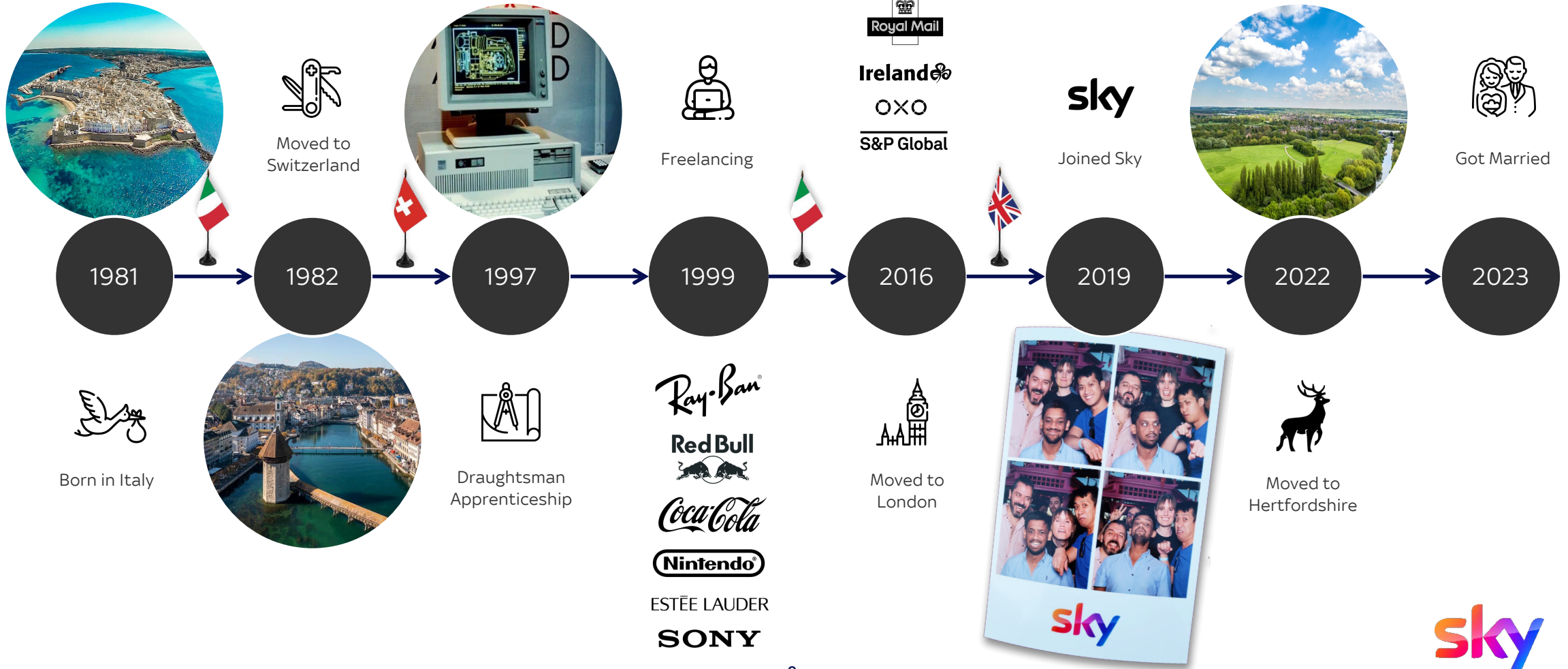
Foodie, Globetrotter, Movies & Videogames Aficionado, Passionate about Tech.

Health conscious. Devoted Husband.

Leading and inspiring the NAI Software Engineering Team in Sky Group Comms

Emanuele Manco

My Life Timeline

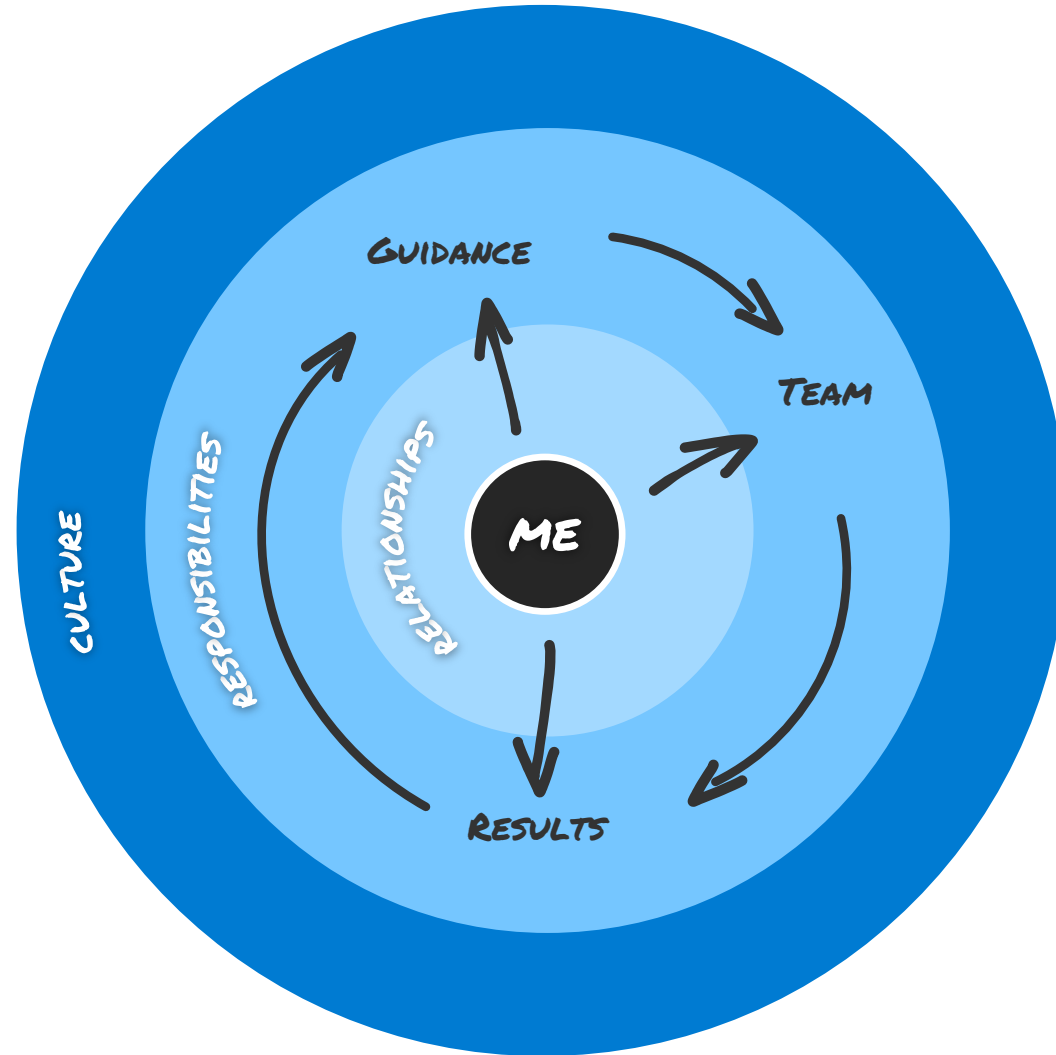


“Culture eats Strategy for Breakfast”

Peter Drucker

Our Culture

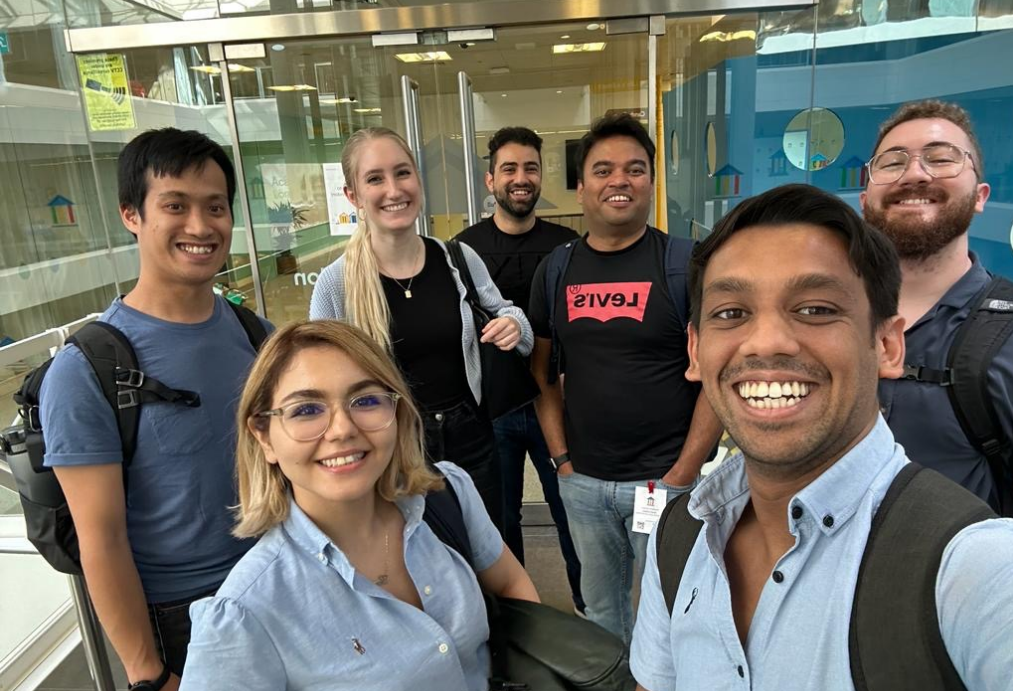
Values & Principles



Source: Radical Candor by Kim Scott







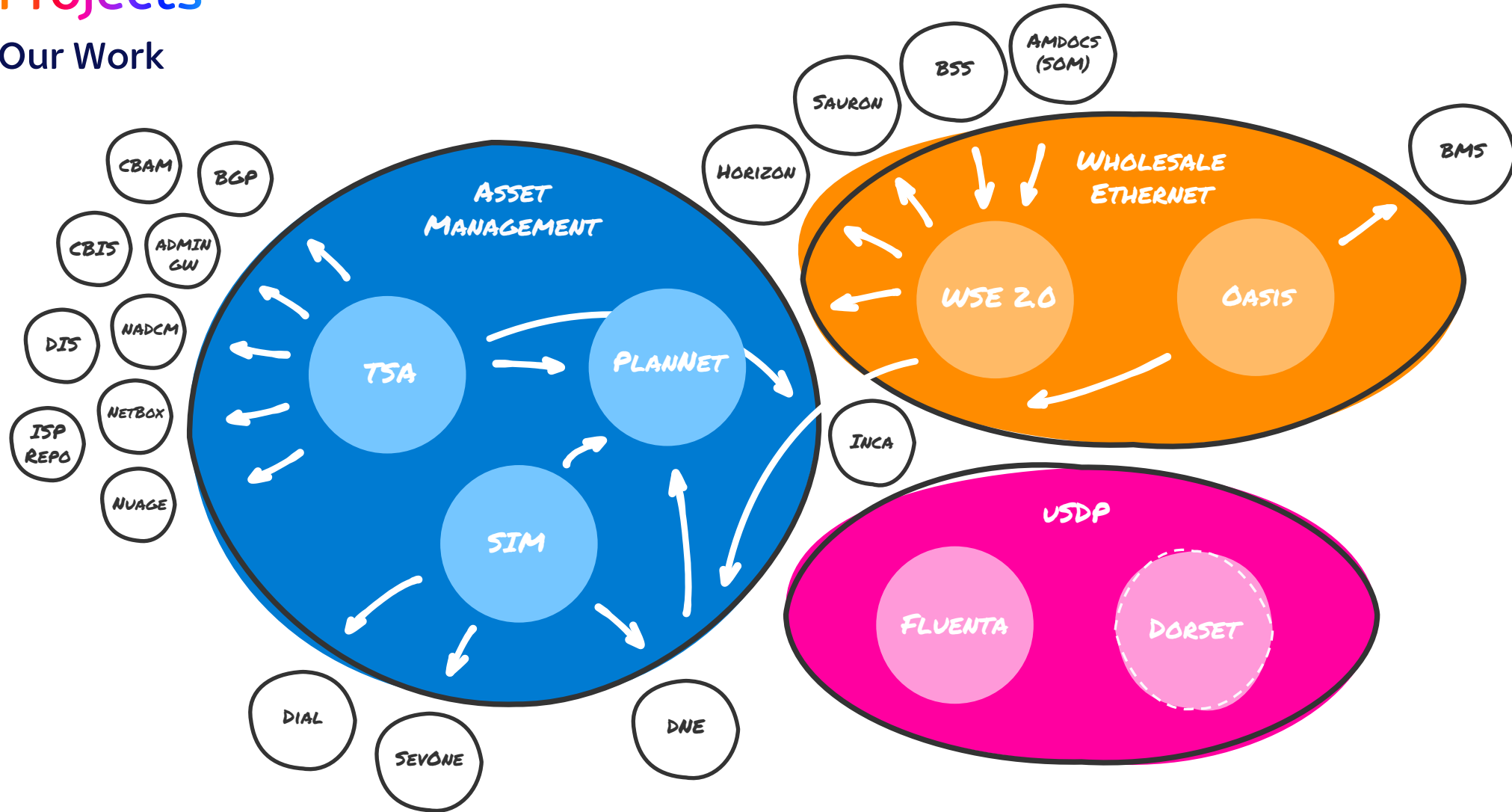


Projects

What we're working on

Projects

Our Work



Agile Methodology

Embracing Change

- At the heart of our development philosophy lies the Agile Methodology, a dynamic framework that enables us to thrive in an ever-changing landscape.
- Agile isn't just a methodology for us; it's a mindset, a culture, and a way of life.
- Through Agile, we embrace change as a constant, allowing us to respond swiftly to evolving requirements and dynamics.
- Collaboration, transparency, and adaptability are the cornerstones of our Agile practices, empowering us to deliver value to our stakeholders at every iteration.
- We've embarked on a journey of Agile transformation, recently adopting SAFe (Scaled Agile Framework) to scale our Agile practices across the organization seamlessly.

Scaled Agile Framework (SAFe)

Elevating our Workflow



What is SAFe?

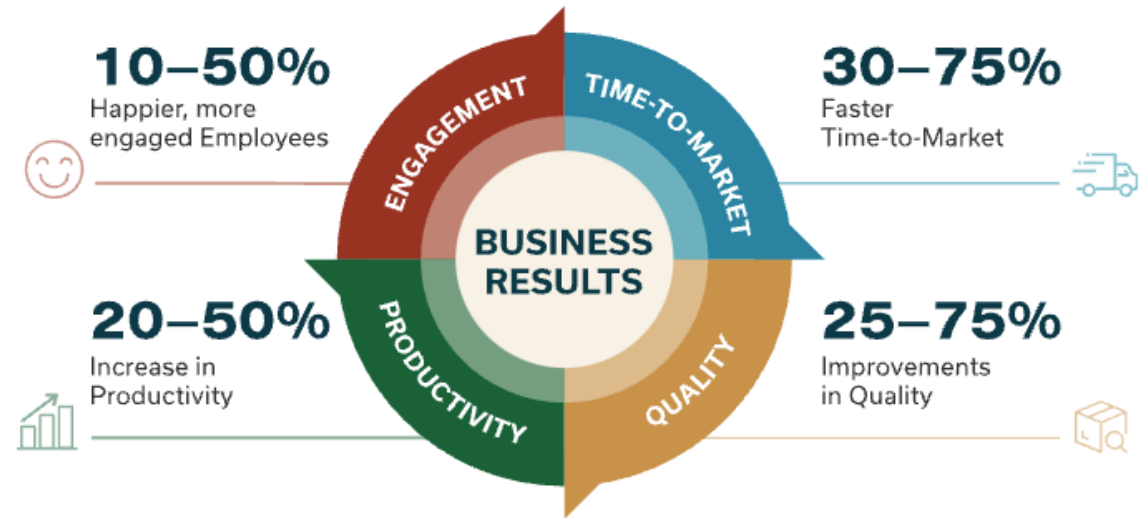
SAFe (Scaled Agile Framework) is a comprehensive framework for implementing agile practices at scale within organisations. It provides guidance on roles, responsibilities, processes, and ceremonies to coordinate and align multiple agile teams working on large-scale projects

Why the need for SAFe

SAFe adoption becomes necessary when an organisation faces challenges in coordinating Agile practices across multiple teams, departments, or projects, requiring a structured framework for scalability and alignment.

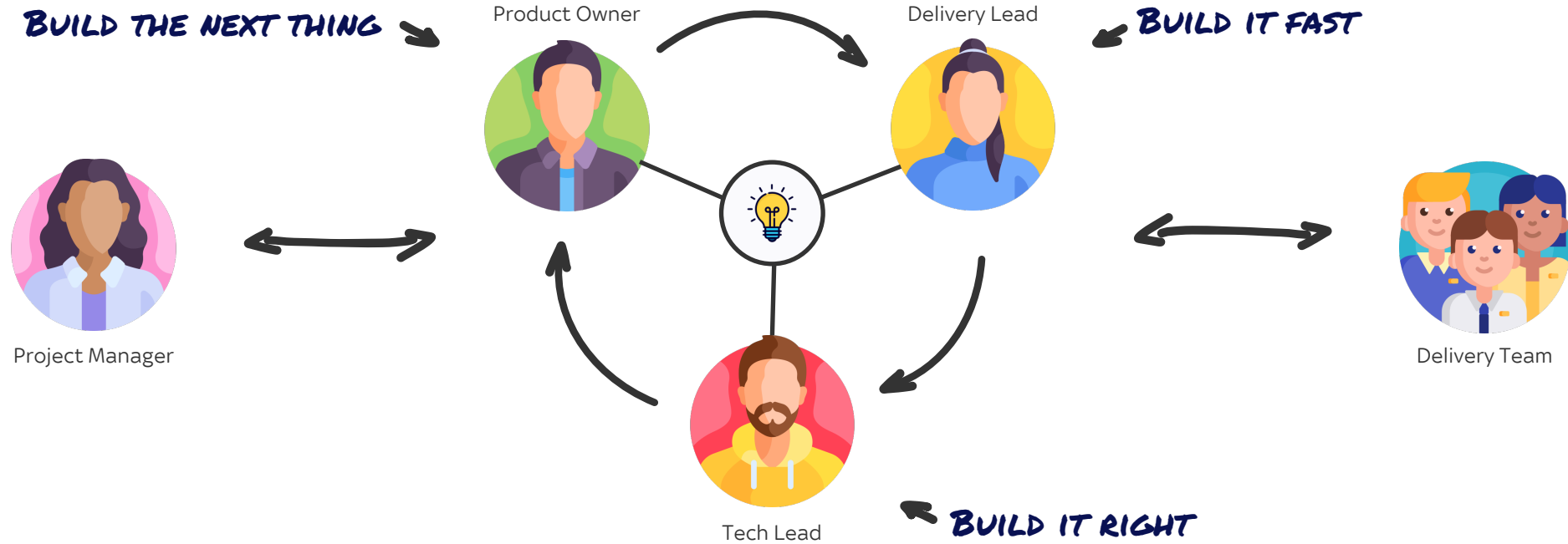
Benefits of SAFe

SAFe drives alignment, communication, and accelerated value delivery through its structured agile framework. It fosters efficiency, risk management, and continuous improvement for enhanced productivity and superior business outcomes.



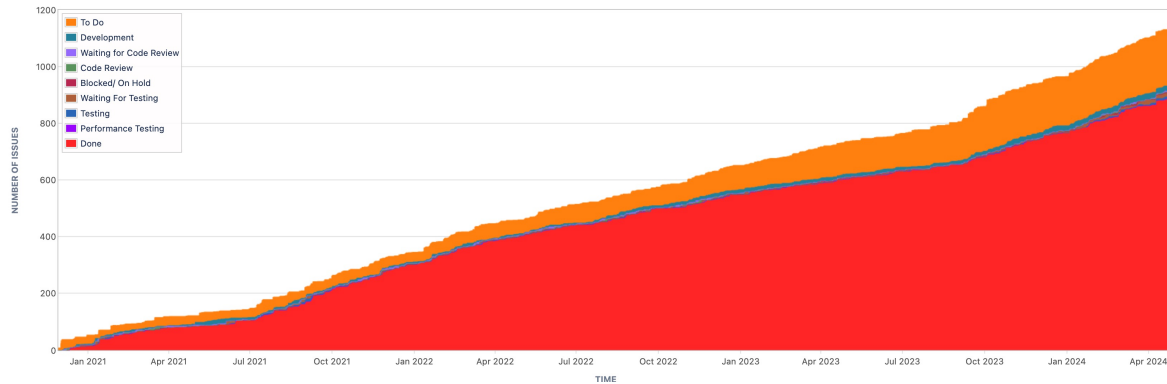
Our Workflow

The process



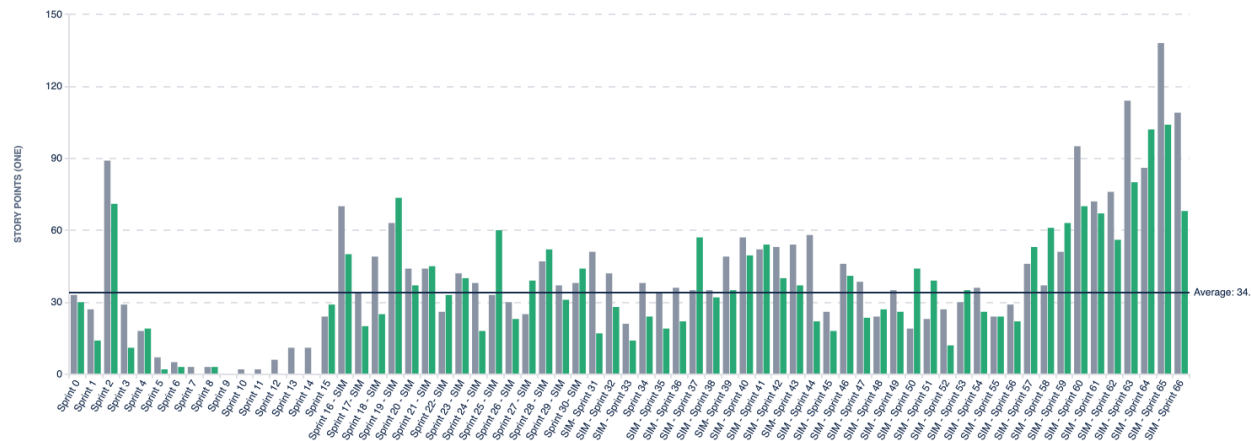
Productivity Metrics

Project Case Study



Cumulative Flow

- A steady increase in completed tasks
- Proportionate increase of “To do” and “Development” * More Feature Requests
- Code Reviews stayed the same
- Steeper growth after we scaled the team



Velocity

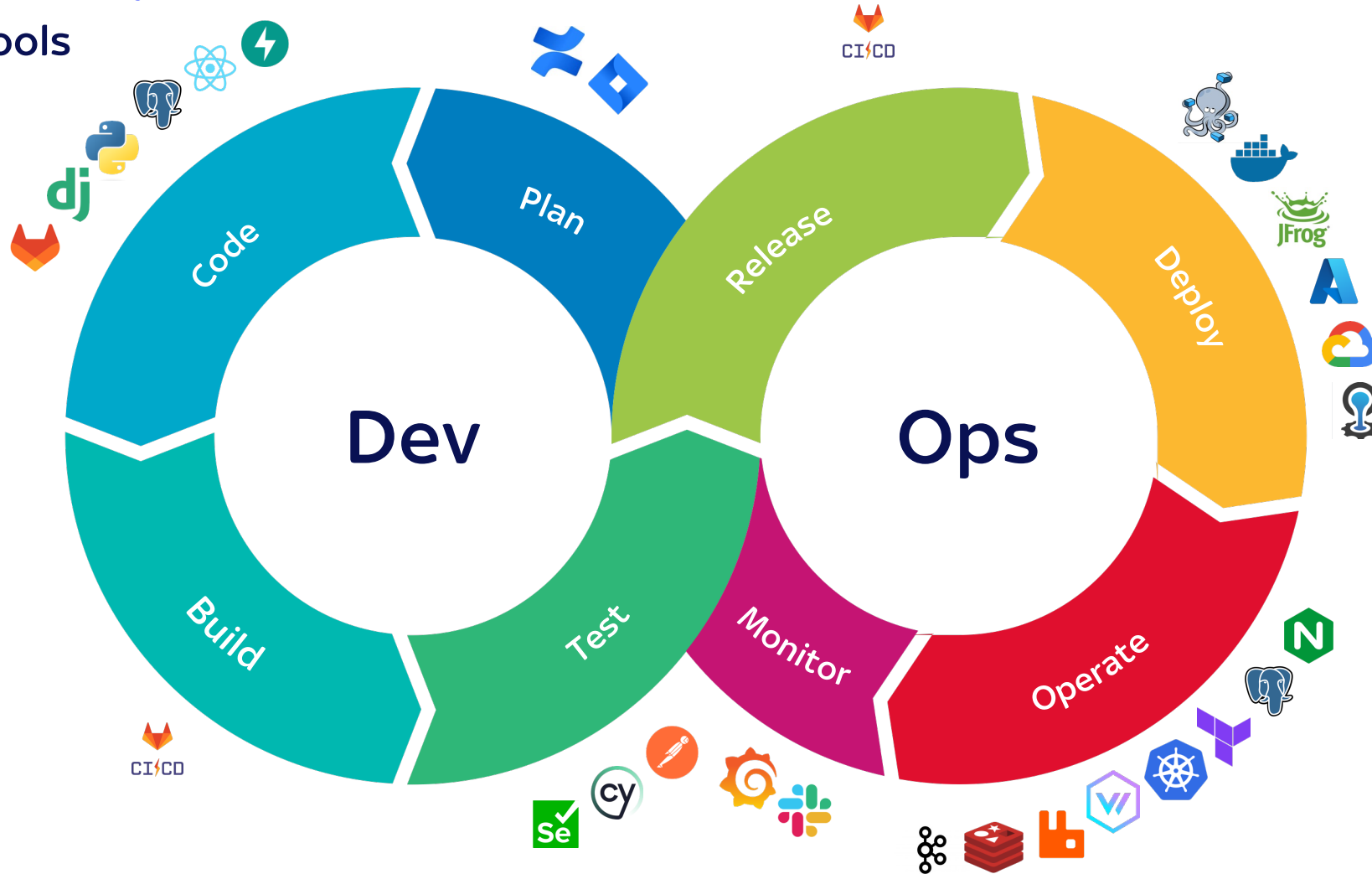
- Scaled the team last year
- Increased Story Points per Sprint from 30 to 120
- Improved Productivity by 300%

Source: JIRA Board



DevOps Lifecycle

Process & Tools



BDD/TDD Approach

The best of both worlds

Behavioral Driven Development

- BDD focuses on the behavior and outcomes of the software from a user's perspective.
- Collaboration between stakeholders, developers, and testers to define and refine requirements.
- Use of a ubiquitous language (often in the form of user stories or scenarios) to bridge the communication gap.
- Scenarios are written using the *Given-When-Then* format to describe the expected behavior.
- Automated tests are created based on these scenarios to validate the software's behavior.
- BDD promotes a shared understanding of requirements and encourages collaboration throughout the development process.

Test Driven Development

- TDD is a development approach where tests are written before writing the actual code.
- Developers start by writing a failing test that describes the desired behavior or functionality.
- Minimal code is then written to pass the test while ensuring it meets the requirements. Once the test passes, the code is refactored to improve its design and maintainability.
- This process is repeated for each new piece of functionality, ensuring that code is thoroughly tested. TDD promotes code quality, as it encourages modular and testable code design.
- It provides a safety net for code changes, ensuring that existing functionality remains intact.

Test Automation

Tools and Frameworks

Oasis

- JavaScript framework
- UI testing with Cypress
- API functional testing with Postman API
- API load testing with K6

Asset Management

- Selenium (Java)
- UI Testing with Cypress

Fluenta

- Python scripts
- Regression testing (Local/CI Pipeline)
- Performance testing (Local)

The collage displays various test automation components:

- Oasis CI Pipeline:** Shows a successful build and test run for 'FP-336 Updated docs' and 'FP-336 Added end to end unit tests'. The pipeline status is 'Passed'.
- Functional Testing (Partner User) on Oasis:** A dashboard showing test results for 'Better business' on 'Wednesday, 12 July 2023 10:06:16'. It reports 6 total iterations, 36 total assertions, 0 total failed tests, and 0 total skipped tests.
- Performance Testing for Fluenta:** A dashboard comparing benchmark and current performance metrics.

Benchmark info				#Fluenta budget: fb_2024_v2.txt			
Version	Commit	Date	Nodes	Interfaces	SRLG Circuits	Card Circuits	Demand Traffic
2.0.1	c37655e	11 Oct 23	594	1760	3488	0	86912
Current info							
2.0.1	162464c	12-Oct-24					
Type	Benchmark	Current	Percentage difference				
Normal Traffic	15,06362	17,64678	-15.8%				
Node Failure	355,98118	202,71832	54.9%				
Trig Failure	308,53511	233,82872	27.5%				
- Elapsed time by failures:** A bar chart comparing 'Normal Traffic', 'Node failure', and 'Trig failure' across 'Benchmark' and 'Current' states.

Failure Type	Benchmark	Current
Normal Traffic	15,06362	17,64678
Node failure	355,98118	202,71832
Trig failure	308,53511	233,82872

Conclusion

Thank you