

# Transforming NRC Data Management for Humanitarian Aid Delivery

The Norwegian Refugee Council (NRC) enhanced its data pipeline to improve error handling, data validation, and automated testing, enabling quicker and more effective humanitarian aid delivery across various regions.



The Norwegian Refugee Council (NRC) is an independent humanitarian organisation working to protect the rights of displaced and affected people during crises. NRC provides assistance to meet immediate humanitarian needs, prevent further displacement and contribute to durable solutions. One of the programs NRC leads is multi-purpose cash assistance to the most vulnerable households affected by the conflict in Ukraine. Improving the data pipeline supporting the open application multi-purpose cash assistance would help improve aid distribution in not only Ukraine, but other regions as well.

## Introduction

In areas hit by war or natural disasters, humanitarian organizations have a tough job: they need to deliver help quickly and effectively to those in need. NRC is working in several regions alongside other NGOs to provide cash assistance to the most vulnerable households so they can cover expenses including rent and health services.

# The Challenge

NRC's current data pipeline used in NRC's Ukraine relief efforts is full of manual steps which can lead to errors. It involves a time-consuming process of downloading files, and manually running a series of scripts. The scripts were developed specifically for the Ukraine cash team and would require modification to use in other country's cash relief programs. The scripts are written in the R language and NRC wanted to convert them to Python to align with their data engineering best practices, improve scalability, enhance maintainability, and integrate better with modern data pipelines and cloud infrastructure.

As a first and critical step in NRC's journey to create a reusable data platform, Sahaj consultants worked with the NRC's tech team to reimplement the Ukraine R scripts in Python which the NRC team knows well. This also created an opportunity to improve reliability by applying best practices such as refactoring common functionality into libraries, an automated test harness and improved error and exception handling. The team also created a 'main' script to automate the sequential execution of the individual stages of the pipeline.

The current system was challenging to handle, expand, and adjust for different emergencies. While it was designed to follow a standard procedure that includes registration, determining eligibility, delivering services, and measuring impact, the system often suffered from inefficiencies and lacked the flexibility needed for future improvements.

The main issue was that the existing system struggled to scale and adapt quickly to meet the varying humanitarian needs across different regions. NRC had two key goals:

- 1 Stabilize the Data Pipeline:** Move from R to Python to make it more supportable, less manual, and more reliable.
- 2 Create a Flexible Data Pipeline:** Develop an automated system that could easily be adjusted for different regions and types of crises, allowing for quicker and more efficient aid delivery in emergencies.

To achieve these goals, NRC partnered with Sahaj Software to create a detailed plan for improving their data management pipeline.

## Solution

The solution involved a multi-phase approach, with a particular focus on migrating the existing R scripts to Python, improving the system's scalability, and setting it up for future flexibility. We also focused on building NRC's internal capacity to manage and extend the pipeline as needed.

# Phase 1: Assessment and Planning

The first step was a detailed assessment of the existing data pipeline. This involved reviewing the R scripts to understand the system's current architecture, limitations, and challenges. The goal was to identify pain points, such as inefficient data handling, lack of error handling, and difficulties with changing requirements or rolling out the data pipeline in new regions. We worked with NRC's technical team to understand their specific needs, including:

- The type of data collected at each stage of the process (e.g., registration, eligibility, service delivery, and impact measurement).
- The infrastructure used to collect and store this data.
- The future goals for the data pipeline, including expansion to other regions and the ability to quickly modify the pipeline.

NRC's long-term goal of creating a template-driven approach for different regions was taken into account when planning the solution. We recommended breaking the project down into manageable phases, each with incremental steps to ensure steady progress.

# Phase 2: Migration to Python

The next major step was migrating the data pipeline from R to Python. This was crucial for stabilizing the system. R was initially selected for its ability to handle data, but NRC required a solution that offered greater flexibility and scalability for handling intricate data tasks and smoother integration with various systems. We led the migration effort, ensuring that the Python code would be modular, maintainable, and aligned with best practices for future enhancements. The Python-based solution provided several key benefits:

- **Scalability**  
Python offered better support for scaling the data pipeline as the program expanded its reach and needed to process more data from a growing number of regions.
- **Error handling and testing**  
Python's rich ecosystem of libraries allowed us to implement more robust error handling, data validation, and automated testing—areas where the original R scripts fell short.
- **Integration**  
Python will allow better integration with NRC's other systems, such as database management tools, data collection interfaces (e.g., mobile apps), and reporting systems.

We also worked closely with NRC's technical team during the migration, explaining the code structure and ensuring they understood the new approach. This was essential for the long-term success of the project, as NRC's team needed to be capable of maintaining and

Once the core pipeline was stabilized through the migration to Python, we helped NRC plan the next stages of the work: developing a generic and automated data pipeline that can be quickly adapted for different humanitarian needs. This involved creating a flexible architecture that could easily accommodate different types of aid programs and regions. We worked with NRC to chart their future path with the following steps:

- Define a modular approach: The pipeline was broken into modular components that could be reused or adapted depending on the specific needs of a given program.
- Create a template-based system: We built a flexible system where the overall pipeline (registration, eligibility determination, service delivery, and impact measurement) could be reconfigured by modifying a set of parameters or templates. This made it possible to quickly redeploy the same pipeline in different regions.

## Phase 3: Testing and Validation

We implemented a robust testing strategy to ensure the system's stability and reliability. Our testing approach included:

- Unit testing: Ensuring that each component of the system worked as expected.
- Integration testing: Verifying that all parts of the pipeline worked together seamlessly.

We achieved 98% test coverage for the new Python codebase, significantly improving the system's reliability and reducing the risk of errors during execution.

## Phase 4: Training and Knowledge Transfer

A key part of the project was ensuring that NRC's team could manage and extend the new pipeline independently. We provided code documentation, and practical guidance on how to modify the system for new regions. This was a critical component of the solution, as we collaborated closely with NRC to ensure long-term self-sufficiency. Together, we established best practices for maintaining the pipeline, equipping NRC's team with the necessary tools and knowledge to independently adapt and enhance the system as new challenges emerged.

## Road Ahead

The migration of the data pipeline from R to Python has the potential to have a significant impact on the operations of cash assistance at NRC in future.

- The modular nature of the new Python-based pipeline can improve the efficiency of data processing and enable NRC to deploy the pipeline to new regions.
- Through close collaboration, NRC's tech team adopted new engineering practices during the training and knowledge transfer process. With these new skills, they can now manage and modify the data pipeline independently, ensuring its long-term sustainability as NRC expands its operations and adapts to evolving humanitarian needs.