

> scalac

best scala hAkkers!

compute  cycles

ComputeCycles

a case study

june 2019



Category:	Minimum Viable Product
Tags:	budgeting, controlling, reporting, data analysis
Industry:	Non-governmental organizations, United Nations
Client website:	computecycles.com

Challenge

Compute Cycles works with NGOs and development agencies like the United Nations. The challenge, and the primary goal of our cooperation, was to build an MVP application from scratch. The software was supposed to be used by governmental institutions to track budgets and value generated by investing in improvement of living conditions or other positive actions in different regions of the world, particularly in Africa.

We had to keep in mind that our end users had different knowledge of the system, from Program Managers who knew which indicators were most important, to people in the field somewhere in Africa (Implementation Managers) who were focused on implementing specific activities, for example drilling new wells.

Based on various business factors, we had to include the following functionalities:

- ▶ To define country-specific programs
- ▶ To monitor and visualize budgets and control expenses in programs, by setting targets, results per countries/provinces in defined periods, and also in 2D and 3D graphs
- ▶ To monitor and visualize the progress of specific actions
- ▶ To maintain various roles in an MVP application for limiting access to data and the application's features according to the user's rights - Implementing Partners, Regional and Program Managers, System Administrators
- ▶ To integrate the payment processes with PayPal procedures
- ▶ To establish and manage the process of CI/CD on a dedicated distributed cloud platform

Solution

Scalac supported ComputeCycles at the following stages:

Stage 1 - Analysis (foundation for development):

- ▶ Gathering **business requirements** and translating them into particular **Epics** and **User Stories** to build the desired **Product Backlog**.
- ▶ Preparing **wireframes** to speed up the process of **designing** crucial elements of GUI.

Stage 2 - Development:

- ▶ Scalac applied an **Agile approach** to managing the process of building the increments of the MVP application. We included **all of the existing Agile principles** and **introduced additional ones** to facilitate the Sprint Review process.
- ▶ Building an MVP application from scratch, our team used the following technologies:
 - Meteor
 - Node.js
 - React
 - D3.js
 - Deck.gl
 - MongoDB
 - OLAP Cubes (Data Cubes)



- ▶ The stack choice was driven by a scalable solution and provided overall flexibility to the system development.
- ▶ The solution was based on **dashboards** and **reports** to ensure the best visualizations of the ongoing processes.

Stage 3 - CI/CD process:

- ▶ Scalac designed and implemented a **complex solution** for Continuous Integration and Continuous Deployment processes. This guaranteed a more stable and better product.
- ▶ Tech stack involved in the solution:
 - GitHub
 - Jenkins
 - Docker Hub
 - Ansible
 - DigitalOcean



Jenkins



ANSIBLE



Results

The released **application is the perfect answer to all of the business needs that were defined in the specifications. The app not only simplifies management**, but also allows control of the budget, which is crucial in every money to value system. Finally, by identifying the needs and distributing funds accordingly, the application meets its ultimate goal - to make **all of the life-improving actions undertaken by the organization happen**.

1. CREATING PROGRAM



2. BUDEGTING

USERS
└ MANAGERS



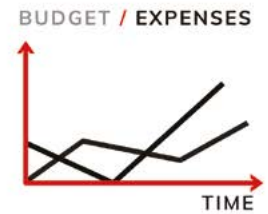
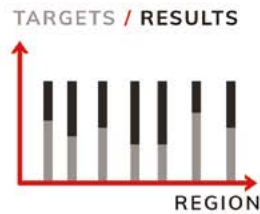
3. COLLECTING RESULTS

USERS
└ REGIONS
└ IMPLEMENTING PARTNERS



4. ANALYSIS

USERS
└ MANAGERS



In line with Agile methodologies, we worked closely with ComputerCycles for several months. Our team included:

▶ **Development Engineers:**

- Three experienced FrontEnd engineers, with BackEnd abilities - who were able to quickly follow technological directions in a truly agile way
- They laid the solid ground for implementation in a fast-paced manner

▶ **QA Engineer:**

One reliable QA engineer - who was able to verify, and point out both business and UX issues

▶ **DevOps Engineers:**

Two experienced DevOps who designed and implemented the complex CI/CD solution, using something so important for ComputeCycles - a distributed environment

▶ **Project Manager / Scrum Master:**

Responsible for conducting all Scrum ceremonies Maintaining effective communication with ComputeCycles

Scalac **took responsibility for every single step**, to deliver a complete solution:

▶ **Analysis & Design**

Gathering requirements, building a valuable product backlog, and designing UX wireframes

▶ **Development**

- Dividing implementation into particular Phases
- Prioritizing backlogs
- Building an application from scratch using best practices and technologies,

to develop reliable, stable, efficient and well-tested code

- Implementing any improvements that were identified during the QA processes
- Conducting scrum ceremonies on a daily basis
- Showing transparency of the implementation process using Jira

► **Deployment**

Designing and implementing the desired CI/CD environment, paying particular attention to security

While working in Scrum, we participated in Agile meetings such as **sprint planning, daily** and **sprint review**. Furthermore, we initiated **additional meetings** with ComputeCycles, during sprints, to share our progress and **facilitate sprint review meetings**.