ADUS TECHNOLOGIES
AGILE SOFTWARE COMPANY

PRODUCTS · PLATFORMS · OPERATIONS

SaaS in action.

Software-as-a-service products built from an empty page to production. A platform taken over from another team and brought live. Told through the SaaS projects we delivered.

5
SaaS products delivered to production

4 + 1
built from scratch, and one taken over

100+
financial KPIs calculated in one platform

3
platforms in one product: web, iOS, Android

AT A GLANCE

Five SaaS products.
Five different starting points.

No two of these projects began the same way: an idea, a half-built platform, a bill for somebody else's service. What they share is the end: a product in production.

01

A product for one industry, from an empty page.

Project and task management made for telecommunication companies: design, back end, front end, cloud and delivery by one team.

Build & Run · delivered from scratch to production

Multi-tenant · Several regions · Single sign-on

02

A platform that changed hands, and went live.

An ecosystem of six business systems for running a company. We started as architecture advisors; three months later the whole development moved to us.

WebAccountPlus · delivered from takeover to production

Multi-banking · Bookkeeping · 100+ KPIs

03

A platform on which others build applications.

A no-code product: users define their own data and compose pages on screen. A CRM and a project tool were built on it as proof.

No-code SaaS platform · delivered from scratch to production

No-code · Permissions · Multi-tenant

04

A paid outside service, replaced by an own solution.

A bank paid a third party for its insurance advisory software. We built a private portal that belongs to the bank, to lower that cost.

Private insurance portal · delivered from scratch to production

Agent workflow · Insurer interfaces · PDF documents

05

An idea that became a subscription product.

A trading companion for day traders, on web, iOS and Android, with subscriptions sold on the web and in both app stores.

xBratAI · delivered from A to Z

Mobile apps · Subscriptions · Scalable cluster

How to read this booklet

Page 3 compares the five products. From page 4 on, every chapter takes one product: where it started, what was built and what it shows. Pages 10–11 describe how we work and what else we run.

THE FIVE PRODUCTS COMPARED

What a SaaS needs, and where we built it.

Features differ from product to product. The parts below repeat in every one of them, and they decide whether a product can be sold to the second, the tenth and the hundredth customer.

	BUILD & RUN	WEBACCOUNT-PLUS	NO-CODE PLATFORM	INSURANCE PORTAL	XBRATAI
Built from scratch	●		●	●	●
Taken over from another team		●			
Requirements worked out with the customer	●		●	●	
Many customers in one installation	●	●	●		
Single sign-on	●	●			●
Interfaces to outside systems	●	●		●	●
Automated tests	●	●			●
Automated build and delivery	●	●	●	●	●
Rule engine		●	●		
Mobile apps and store subscriptions					●
Where it runs	Google Cloud	Azure	On-premise	On-premise	Own cluster
Result	In production	In production	In production	In production	In production

A dot marks what the project documents state. An empty field means “not part of this project or not documented”, not “impossible”.

3 months
from advisory contract to taking over a whole development

6
business systems connected in one ecosystem

100+
questions in one insurance questionnaire, turned into documents

1,730
automated tests guard one product

What repeats: every product has its own automated build and delivery, and each one runs where the customer needs it: at a large cloud provider, in the customer's own data centre or on a cluster we built for it.

PRODUCT 01 · BUILD & RUN · WWW.BUILDANDRUN.COM

Project management for telecom, from an empty page to production.

Build & Run is a project and task management system adjusted to the needs of the telecommunication industry. There was no earlier version: we designed it, built it and brought it to production.

Specify
Requirements consulted and sharpened with the customer

Design
Screens and user flows drawn before coding

Build
Back end and front end written from scratch

Automate
Containers and a delivery pipeline

Run
Tested version delivered to production

What makes it a SaaS, not just an application

Many customers, one system

The architecture follows the multi-tenancy model: every customer company works in its own space of the same installation.

Business value: a new customer is a setting, not a new installation

Ready for several regions

The infrastructure was designed, documented and built to scale across geographic locations in Google Cloud.

Business value: the product can follow its customers abroad

Sign-in people already have

Users log in with their Google or Microsoft account. No extra password to create, forget and reset.

Business value: less friction on the first day

Files stay with the customer

Documents are stored in the Google Drive or OneDrive of the customer company, not in a storage of the product.

Business value: the customer keeps control of its own files

Our part

- [x] User interface and experience designs
- [x] Contribution to the specification
- [x] Back end and front end, from scratch
- [x] Containers for every part of the system
- [x] Continuous integration and delivery pipelines
- [x] Cloud infrastructure: design, documentation, build
- [x] Delivery of the tested version to production

Technology

Front end	Angular
Back end	Java, Spring, JUnit
Database	PostgreSQL
Design	Figma
Operations	Docker, Drone CI, Google Cloud Platform

What this project shows

A software company with an idea for its own industry got a finished product without building a product team first.

PRODUCT 02 · WEBACCOUNTPLUS · WEBACCOUNTPLUS.SWISS

Six business systems, one ecosystem, taken over and brought live.

WebAccountPlus connects the services a company needs to run its business. We were hired to advise on the architecture. After three months the customer moved the whole development from the previous team to us.

CRM
customers and contacts

EDI
document scanning and recognition

Asset management
what the company owns

BI analytics
data and reports

Bookkeeping
accounting software

Digital Corporate Advisor
banking, payments, master data

Start
Architecture advisory only

Month 3
Whole development moved to us

Build
Banking, payments, sign-on, cloud

Live
Delivered from takeover to production

Phase 2
Advisory screens and rule engine

What we added after the takeover

All banks in one place

Multi-banking over the ISO 20022 standard, through the EBICS and b-Link channels: balances, credit and debit transactions, payments.

One view of the money in every bank

Payments that find their invoice

Invoices and expenses from the other systems are paired with bank transactions from end-of-day and intraday reports. Outgoing payments are sent on the due date.

Less manual matching

Straight into the books

Documents and payments are booked into Abacus, the most widely used bookkeeping software in Switzerland.

No second entry of the same document

One login for everything

A single sign-on server connects the systems of the ecosystem, so the user moves between them without logging in again.

Six systems feel like one

One owner, several companies

A user who owns more than one company sees charts and overviews of all of them at once.

The whole group on one screen

Figures in real time

A complete balance sheet overview, always current, and more than 100 financial KPIs calculated from the data.

A basis for advice, not only for records

6
systems with defined roles and interfaces

3 months
from advisory to full takeover

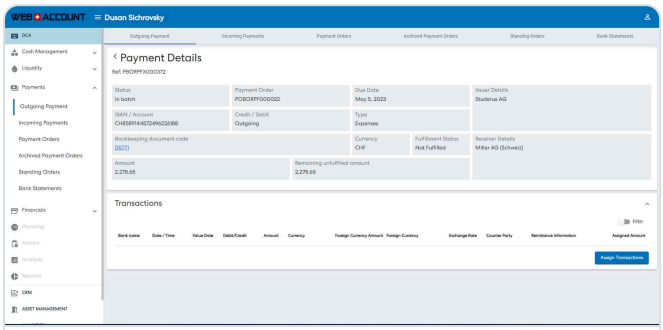
100+
financial KPI calculations

4
environments in Azure: development, test, staging, production

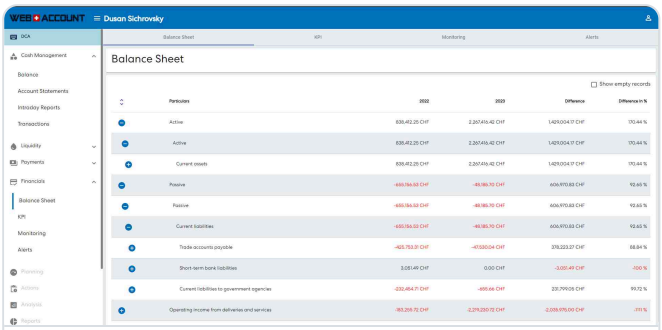
PRODUCT 02 · A LOOK INSIDE

WebAccountPlus on screen.

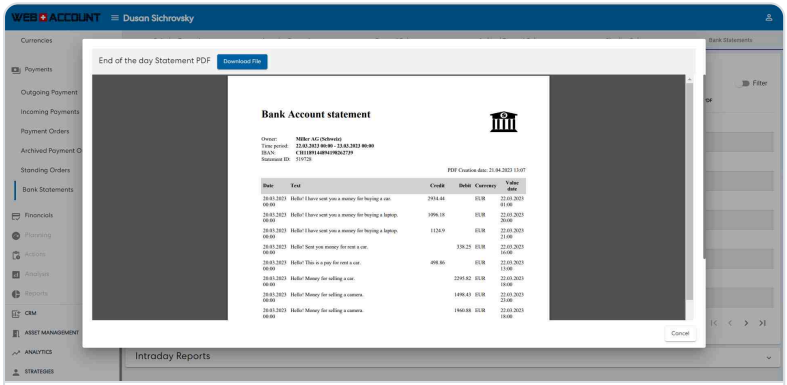
Three screens of the Digital Corporate Advisor: the part of the ecosystem that talks to the banks and feeds the bookkeeping.



Balance sheet. Two years side by side, with the difference per line.



Payment details. One payment order with its bank transactions.



Bank statement. The end-of-day statement, as received from the bank.

Quality behind the screens

- Unit tests cover the back end
- Integration tests run whole business processes across all systems
- Separate environments for development, test, staging and production
- Operation and maintenance of the ecosystem in the cloud

Second phase

- Screens for bank advisory and financial advisory
- A rule engine that turns KPIs into actions
- Evaluation criteria the customer can set up

What this project shows

A platform does not have to be ours from the first line. We can enter as advisors, take over the development and carry it to production.

Technology

Front end	Angular
Back end	Java, Spring, JUnit
Database	MS SQL
Testing	JUnit, Playwright
Banking	ISO 20022, EBICS, b-Link
Sign-on	OpenID Connect, OAuth2, SAML
Operations	Docker, Azure pipelines, Azure cloud

PRODUCT 03 · MULTI-PURPOSE NO-CODE SAAS PLATFORM

A platform on which applications are composed, not coded.

Users of this platform define their own data model and build pages and layouts on screen, grouped into applications. It supports almost any business scenario with a wide set of interface components.

Data model defined by the user

Tables, fields and relations are created in the product itself, while it runs.

A new business need does not wait for a release

Pages built on screen

Layouts and pages are composed in a what-you-see-is-what-you-get editor.

The person who knows the process builds the screen

Users and permissions

An advanced management of users and rights decides who sees and changes what.

One platform for many roles

Files and folders

File management with a folder structure belongs to the platform.

Documents next to the data they belong to

Many customers

Multi-tenancy support: each customer has its own applications and data.

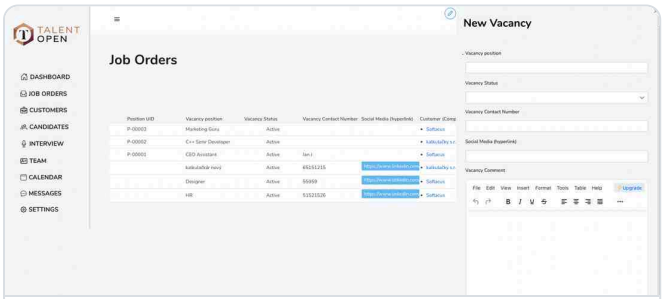
One installation, many businesses

Applications as proof

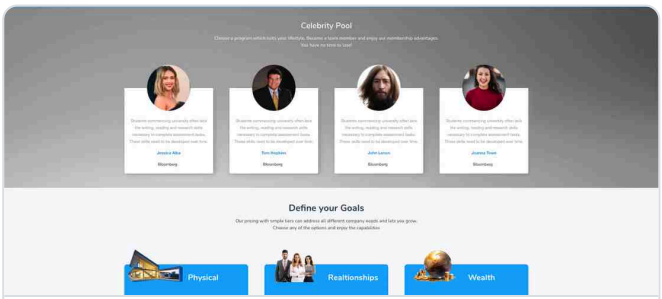
A CRM and a project management tool were created inside the platform; a rule engine and an application store follow.

The platform proves itself on real applications

Two applications built on the platform



Recruiting application. Job orders, candidates, interviews and a calendar.



Public pages. Rendered on the server, so they load fast and can be found by search engines.

Our part

- Requirements, processes and workflows worked out with the customer
- Architecture of the SaaS, with emphasis on performance
- On-premise infrastructure with several containers
- Automated build and delivery
- Delivered from scratch to production

Technology

Front end	Angular, with server-side rendering
Back end	Node.js (TypeScript)
Database	MongoDB
Operations	Docker Compose, Drone CI, on-premise

PRODUCT 04 · PRIVATE INSURANCE PORTAL · MORTGAGE BANK, AUSTRIA

From a paid outside service to a portal the bank owns.

A bank used a paid third-party SaaS for its insurance advisory. The aim of the project was to lower that cost by moving to a private solution. We built the insurance portal for the needs of the bank and fitted it into the portal it already had.

Before a subscription paid to an outside provider → **After** an own portal inside the bank's solution

Before no experience with software design on the customer side → **After** requirements, processes and workflows written down

What was built

The agent's workflow

An advanced workflow leads the insurance agent through the advisory, step by step.

The same quality of advice every time

Connected insurers

The portal talks to the systems of outside insurance companies through their interfaces.

Offers without retyping

Documents from answers

A generator fills PDF templates from an insurance questionnaire with more than 100 questions.

A finished document at the end of the talk

100+ questions in the insurance questionnaire

10 × less programming logic through a questionnaire framework

0 customer infrastructure needed for development: it ran in our own cloud

1 portal for the user: the new part fits the existing one

Our part

- Requirements, processes and workflows with the customer
- Portal architecture that fits the existing portal
- User experience design
- Development environment in the ADUS on-premise cloud
- Automated build and delivery
- Delivered from scratch to production

Technology

Front end	Vue.js
Back end	.NET C#
Database	MS SQL
Interfaces	SOAP, JSON
Operations	Docker running Mono, TeamCity CI, on-premise

What this project shows

A SaaS subscription is not always the cheapest way. When a service is central to the business, owning it can pay off.

PRODUCT 05 · XBRATAI · XBRATAI.COM

An idea in spring.
A subscription product on three platforms.

xBratAI is a trading companion for day traders: it watches the markets around the clock and notifies its subscribers the moment all conditions of a trading method align. The method belongs to the founder. The application — back end, web app, mobile apps and cloud platform — was built by ADUS Technologies.

4
markets: forex, crypto, futures, stocks

55
instruments watched in real time

3
platforms: web, iOS, Android

11
subscription plans, monthly and yearly

What a subscriber gets, and what had to be built for it

REAL TIME

The signal, the second it happens

Entry point, direction and live updates arrive as a push notification and on the open screen at the same moment.

RELIABILITY

No signal may get lost

Every incoming signal is stored before it is processed and can be replayed. A failed step waits in a queue.

REVENUE

Subscriptions on three channels

Web checkout, Apple App Store and Google Play purchases end in one account, checked on the server.

PROTECTION

The signals are the product

Access to the live feed is guarded per session, and marked signals show where a leak came from.

ENGAGEMENT

More than alerts

Trading records and statistics, a moderated community and an academy keep subscribers in the product.

BACK OFFICE

The operator is in control

The product team manages users, roles, signal types and settings itself, with an audit trail.

Built to scale and to stay up

- ☑ **Grows by itself.** Each service runs in 2 to 10 copies; the cluster adds servers from 3 to 10 when demand rises.
- ☑ **No single point of failure.** Databases and the message queue run in three copies each.
- ☑ **Infrastructure as code.** The whole platform can be rebuilt from the repository.
- ☑ **Releases as routine.** Four environments from test to production; new versions roll out automatically.
- ☑ **No lock-in.** Open technology that is not tied to one cloud vendor.

1,730
automated tests guard back end and mobile app

19 months
from first commit to version 2.1.1 in both stores

130k+
lines of code in back end, web and mobile app

5
services scaled separately, joined by a message queue

Product figures as published on xbratai.com, November 2026. xBratAI is a product of xBrat Mobile Technologies FZCO, Dubai. Trading results are not part of our delivery and are not quoted here.

HOW WE BUILD A SAAS

From the first talk to the first paying user — and after.

The steps are simple. What makes the difference is that nothing is handed over a wall: the people who design the product also build it, release it and answer the phone when it runs.

STEP 1

Understand

We start with your business and your users, not with technology. Result: a short written scope and the first budget frame.

STEP 2

Design

Clickable screens before any code. You change your mind here, where it costs nothing.

STEP 3

Plan the first release

The smallest product that already earns or saves money. Everything else goes on the list for later.

STEP 4

Build in short rounds

After every round you see working software and decide what comes next. You have access to our task board.

STEP 5

Test automatically

Tests are written with the code. Before each release the whole product is checked by machine.

STEP 6

Launch

Cloud environment, store listings for mobile apps, payment channels, monitoring — all in place before the first user.

STEP 7

Operate

We watch the system, apply updates, keep backups and react to incidents.

STEP 8

Grow

New features, new markets, more users. The architecture was chosen for this from the start.

What a SaaS needs besides its features

The list founders usually discover after launch. We plan it before.

- [x] Each customer's data separated from the others
- [x] Sign-in with company or social accounts
- [x] Plans, trials, renewals, cancellations — on web and in the app stores
- [x] Roles and an administration area for your own team
- [x] Notifications that arrive: push, e-mail, in the app
- [x] Capacity that follows demand, up and down
- [x] Backups that have been restored at least once
- [x] A record of who changed what, and when
- [x] Protection of the content you sell
- [x] Releases that users do not notice

Who works on it



Typical team in our product projects: one lead, two to seven developers, one operations engineer. One contact person for you throughout.

Three ways to start

- **From scratch:** an idea and a budget frame are enough.
- **By takeover:** we begin as advisors next to your current team.
- **By replacement:** an own solution instead of a subscription you pay to somebody else.

OTHER PROJECTS WORTH MENTIONING

The same team keeps large business systems running.

A SaaS product has to run every day, for years. We know what that takes, because next to our product work we maintain the core systems of a large international holding under service level agreements.

700
database tables in a custom ERP for paper traders, in production since 2003

600
database tables in a custom ERP for paper factories

300
database tables in bookkeeping software connected to both ERPs, in production since 2003

250
servers taken over under service level agreement: 150 Windows, 100 Linux

All three systems: Java, Spring, IBM DB2. Server infrastructure taken over from June 2023.

High-tech projects next to the enterprise work

AUTOMOTIVE · MOBILE

The car talks to the tablet

A mobile application connects a car with an Android device and streams the parameters of the car's diagnostic interface in real time.

Live vehicle data in front of the crew

BUILDINGS · POSITIONING

The phone instead of a smart card

A proof of concept for the Quuppa positioning technology: indoor location moved from a proprietary active smart card to the iPhone.

Proof of concept delivered successfully

ENGINEERING DATA · MIGRATION

From IBM Rational DOORS to the next generation

A solution that migrates data from the legacy requirements system to IBM DOORS Next Generation, using the specialised DXL language and analysis of the system's interfaces.

Legacy data carried into the new platform

AUTOMOTIVE · PRODUCTION

Data entry in the manufacturing hall

A portal that digitalises data gathering in an automotive factory: large touch screens placed directly in the halls.

Production data captured where it arises

Why this matters for a SaaS

Building a product is the shorter part of its life. The team that writes your SaaS has years of practice in the longer part: support, releases, servers and response times agreed in a contract.

IS THIS YOUR SITUATION?

One yes is enough to start a conversation.

1

A product idea for your industry, and no product team?

Built from an empty page — page 4

2

A platform half built, and a team that cannot finish it?

Taken over and brought live — pages 5–6

3

Customers who each need their own application?

A platform to compose them — page 7

4

A subscription for a service that is central to your business?

An own solution instead — page 8

5

A product that must live on the web and in both app stores?

From A to Z — page 9

6

A running product that needs operation under contract?

Service level agreements — page 11

What happens after your call

Day 1
A conversation about your situation, with people who have done it

Week 1
We look at the idea or the existing system — under a confidentiality agreement

Week 2
A written proposal: steps, team, time frame, budget frame

Then
A small first step you can judge us by

ADUS Technologies s.r.o.

Joliota Curie 739/18 · 058 01 Poprad · Slovakia, EU

Contact person: Dušan Sichrovský, CEO

www.adus.tech · info@adus.tech · +421 949 407 310