Unique-P

Sample report

Takeover Assessment

Muster AG's B2B ordering portal "Orderdesk": a technical inventory, the risks and a plan for keeping the application running safely and developing it further.

Findings at a glance	As of the report
Overall risk 4 critical, 5 high findings	High
Takeover readiness Conditional: easy to take over with immediate actions	6 / 10
Deadline to act Hosting provider switches off PHP 7.4	March 31, 2027

Client
Muster AG, wholesaler of fastening technology

Application
"Orderdesk" B2B ordering portal (PHP / Laravel)

Full access · Report
September 11, 2026 · September 18, 2026

Written by
Dimitri König, Unique-P GmbH

Sample: The company, application and findings in this report are entirely fictional. Structure, depth and tone match a real Takeover Assessment by Unique-P GmbH.

The application can be taken over. What's critical isn't the code but its surroundings.

"Orderdesk" handles about 55% of Muster AG's revenue: 420 active business customers, about 1,800 orders per month. The application was developed from 2014 onward by an external freelancer who has been unreachable since 2025. The hosting provider has announced it will switch off PHP 7.4 on March 31, 2027.

The core business logic is solidly built and easy to read; a rebuild is not necessary. The risks lie in operations: the platform runs on versions without security updates, there is no test environment and there are no tested backups, and part of the production code is not under version control. Several findings can be fixed right away and should be tackled regardless of any further decision.

Overall risk	Takeover readiness	Deadline to act
High	**6 / 10**	**March 31, 2027**
4 critical, 5 high findings	Conditional: easy to take over with immediate actions	Hosting provider switches off PHP 7.4

The three most important risks

1. **Downtime from April 2027.** Without an upgrade, "Orderdesk" stops running once PHP 7.4 is switched off. The upgrade affects the framework, packages and database at the same time.
2. **Data loss in an outage.** There are only the hosting provider's database backups (7 days). Files are not backed up, and a restore has never been tested.
3. **Publicly accessible customer data.** CSV exports with customer and order data sit in a publicly reachable directory. This can be fixed within a day.

Recommendation: Option B, stabilize and modernize step by step. In three stages within about five months of the decision, so before the shutdown, each at a fixed price set before it starts. A rebuild would cost an estimated CHF 180,000 to 250,000 and would not be finished before the shutdown.

Decision needed: choice of option within three weeks of the report. The immediate actions (chapter 12) should start right away.

02 · Takeover readiness

Can the system be taken over safely?

6 / 10 conditionally ready for takeover

Criterion	Finding	Status
Source code	Fully backed up from the server and reconciled with the code in Git	available
Version history	Git repository until 2021; after that, 31 files were changed only directly on the server	partial
Hosting & domain	Access available, but the hosting account and domain are registered to the developer's personal email address	partial
Database & backups	Access available; only the hosting provider's database backups, restore never tested	partial
Test/staging environment	None. Changes go straight to production	missing
Documentation	README with installation notes from 2016, nothing else	missing
Business knowledge	The head of inside sales knows every process and special case	available
Handover from predecessor	Not possible; knowledge was reconstructed from code, database and operations	missing

03 · Technology inventory

What's in use, and for how much longer.

Component	Version	End of support	Rating
PHP	`7.4.33`	November 2022	critical
Framework	`Laravel 5.8`	February 2020	critical
Database	`MySQL 5.7`	October 2023	high
Frontend	`jQuery 1.12, Vue.js 2.6`	Vue 2: December 2023	high
PHP packages	`64 packages`	23 without security updates	high
PDF generation	`wkhtmltopdf 0.12`	Project archived since 2023	medium
Frontend build	`Laravel Mix, 212 npm packages`	Build not reproducible	medium
Web server	`Apache 2.4 (shared hosting)`	Maintained by the host	low

04 · Lifecycle risks

The calendar has a say.

The runtime, framework and database have not received security updates for years. Until now, that was a silent risk. With the announced shutdown of PHP 7.4, it now has a date.



02/2020	11/2022	10/2023	12/2023	09/2026	03/2027
Laravel 5.8 without updates	PHP 7.4 without updates	MySQL 5.7 without updates	Vue 2 without updates	Assessment	Host switches off PHP 7.4

Important: Buying time with the hosting provider's "legacy PHP" offering only moves the date. The security vulnerabilities remain, and every additional year makes the version jump bigger.

05 · Deployment & operations

How the application is run today.

Deployment

- Upload via FTP from the developer's laptop, including the `vendor/` directory
- No build process on the server, no deployment documentation
- No rollback: a faulty change stays live until someone finds it

Operations

- 11 cron jobs in the hosting panel, 3 of them with no discernible purpose
- No monitoring: customers are the first to notice outages
- Log files without rotation, currently 2.4 GB
- Order confirmations via PHP `mail()`, some of them landing in spam

Silent failure found: Since March 2026, the `prices:sync` cron job has been aborting due to timeouts, without any alert. As a result, about 6% of products show outdated prices.

06 · Codebase

A solid core with a few dangerous shortcuts.

Size	Automated tests	Maintainability
47,500	**0**	**moderate**
lines of PHP and JavaScript	no safety net for changes	clear naming, but core logic concentrated in one place

- **Positive:** consistent Laravel structure, understandable names, business rules mostly in one place. New developers find their way around quickly.
- **Pricing logic concentrated:** The entire price calculation, including special terms, sits in a single class of 2,100 lines. It works, but without tests it is risky to change.
- **Changes in the `vendor/` directory:** Two third-party packages were modified directly. The modifications will be lost with the next package update.
- **Raw SQL queries:** 14 queries are built by string concatenation, two of them with user input (see chapter 08).

07 · Data & integrations

What the application depends on.

Integration	How it works	Finding
ERP · orders	Nightly CSV export via SFTP	No check whether the import into the ERP succeeds. Missing orders are first noticed by the customer.
ERP · products & prices	Daily CSV import	Mapping documented only in the code; partly failing since March 2026 (chapter 05).
Payment provider	Credit card and invoice	Outdated API version; the provider has announced the changeover.
Email	The host's PHP `mail()`	Sender not authenticated, delivery unreliable.

Database: 86 tables, 4.2 GB; largest table `order_items` with 11 million rows. No signs of inconsistent data.

08 · Security warning signs

Obvious risks that stand out right away.

The assessment is not a substitute for a penetration test. Listed here are warning signs that became visible while analyzing the code and operations.

Finding	Severity
CSV exports with customer and order data in a publicly reachable directory	critical
Production credentials (database, SFTP, email) in the configuration file in the Git repository	critical
Two queries with user input vulnerable to SQL injection (search filter, export)	critical
Admin login without protection against password guessing, without two-factor authentication	high
jQuery 1.12 with known XSS vulnerabilities	medium

09 · Knowledge risk

What only the predecessor knew.

- Deployment process and server structure
- Mapping of the ERP fields for export and import
- Special terms for 3 customer groups, hard-coded
- Purpose of the 11 cron jobs
- Access to the hosting account and domain

All of this was reconstructed during the assessment from code, database and operations, and documented in writing.

10 · Business continuity

What happens in an outage.

Scenario	Today	Target
Server goes down	unclear, several days	< 4 hrs
Faulty update	no rollback	< 15 min
ERP export fails	unnoticed	immediate alert
Developer unavailable	standstill	any qualified developer can take over

11 · Risk matrix

All findings at a glance.

No.	Risk	Severity
R1	PHP 7.4 switched off as of March 31, 2027	critical
R2	Publicly accessible customer data	critical
R3	Credentials in the repository	critical
R4	SQL injection in search filter and export	critical
R5	No tested backups	high
R6	Hosting and domain in a private individual's name	high
R7	No monitoring, silent failures	high
R8	Admin login unprotected	high
R9	No test environment, no tests	high
R10	ERP integration without checks	medium
R11	Unversioned changes on the server and in the vendor/ directory	medium
R12	Outdated frontend (jQuery, Vue 2)	medium
R13	PDF generation without maintenance	medium
R14	Unreliable email delivery	low
R15	Log files without rotation	low

	minor	moderate	major
likely	R12	R7 R9	R1 R2 R3 R4
possible	R14 R15	R10 R11 R13	R5 R6 R8
rare			

Rows: likelihood · Columns: impact

Summary

4 critical · 5 high · 4 medium · 2 low

Eight of the nine critical and high risks can be fixed or significantly mitigated within 30 days.

12 · Immediate actions (0–30 days)

Secure things first. Most of it takes hours, not weeks.

No.	Action	Mitigates	When
M1	Block access to the public exports and move them outside the web root	R2	Day 1
M2	Change all credentials, remove them from the repository and store them in the shared password vault	R3	Day 1
M3	Close the two SQL injection holes	R4	Week 1
M4	Daily backups of the database and files to a second location in Switzerland; test and document the restore	R5	Week 1
M5	Transfer the hosting account and domain to Muster AG	R6	Week 1–2
M6	Move the complete production code into a GitHub repository owned by Muster AG, including documentation	R11	Week 2
M7	Monitoring and alerts for availability, errors, cron jobs and the ERP export	R7 R10	Week 2
M8	Admin login with two-factor authentication and protection against password guessing	R8	Week 3
M9	Set up a test environment (staging) with anonymized data	R9	Week 3–4

13 · Recommendations (3–12 months)

Then modernize, while the application keeps running.

- **Safety net:** automated tests for the 12 business-critical workflows (login, price calculation, ordering, ERP export).
- **Upgrade:** PHP 8.4 and the current Laravel version in several intermediate steps, each one tested and shipped on its own.
- **Database:** move to MySQL 8.4 LTS with a trial run on staging.
- **Deployment:** automated via GitHub, with sign-off on staging and one-click rollback.
- **Integrations:** ERP export with validation, logging and alerts; payment provider on the current API.
- **Legacy baggage:** replace wkhtmltopdf, update jQuery, move Vue 2 to Vue 3 step by step.

14 · Strategic options

Four paths. One fits.

A · Keep it running	Immediate actions only; extend PHP 7.4 through a paid legacy offering from the hosting provider. Cheap, but the security vulnerabilities remain and the version jump gets bigger every year.	**not recommended**
B · Stabilize & modernize	Secure first, then upgrade in stages while the application keeps running. The proven business logic is preserved, and the risk drops with every stage. Completed before the shutdown.	**recommended**
C · Replace parts	For example, rebuild the frontend and keep the backend. Only makes sense once new usability requirements come up after stage 3.	**revisit later**
D · Rebuild from scratch	An estimated CHF 180,000 to 250,000 and 9 to 12 months. Not finished before the shutdown, and the special rules that have grown over the years would have to be captured all over again.	**not necessary**

15 · Implementation proposal

Option B in three stages, each at a fixed price.

Stage	Scope	Time from decision	Price
1 · Secure	Immediate actions M1 to M9	Month 1	Fixed price
2 · Safety net	Tests for critical workflows, automated deployment, Laravel up to version 8	Month 2	Fixed price
3 · Modernize	PHP 8.4, current Laravel version, MySQL 8.4, payment API	Months 3 to 5	Fixed price

Each stage has a fixed price that is set before it starts. Completed before PHP 7.4 is switched off.

Support from stage 1: monthly flat fee for monitoring, updates and further development (with a fixed number of hours), response on the next business day, emergencies prioritized.

Next steps

What happens now.

1. **Walkthrough of the report** (60 minutes) with management and inside sales, answering questions.
2. **Decision on the option** within three weeks of the report.
3. **Start of stage 1** in the week after the decision, first actions on day one.

Nobody depends on a single person again

The takeover should end the dependency on a single person, not shift it. That's why this applies from day one:

Code & documentation

Every line of code and all documentation live in Muster AG's GitHub repository. Muster AG has full access at all times.

Credentials

All credentials (hosting, database, domain, third-party services) are kept in a shared password vault that Muster AG can access at any time. Never in the code.

Operations

Deployment, backups and restore are documented as step-by-step guides. Another qualified developer can take over at any time.

Ownership

The hosting account, domain and all licenses are in Muster AG's name. No license rental, no lock-in.



Dimitri König
Unique-P GmbH · hallo@unique-p.ch · +41 76 231 32 71 · unique-p.ch

Sample report: The company, application and findings are fictional. This is what the report you receive after a Takeover Assessment looks like: for your application, with your numbers. Start with the free Fit Check at unique-p.ch/en/services/web-application-takeover.