Cross-Cutting

Bilko cross-cutting concerns — change requests, lessons learned, tech debt

- Change Request Template
- Lessons Learned
- Tech Debt Log

Change Request Template

Change Request Process: Bilko

> **Project:** Bilko — Balkan Accounting SaaS **Version:** 0.1 **Date:** 2026-02-23
> **Author:** John (AI Director) **Status:** Draft **Reviewers:** Alem Bašić (CEO)

Document History

Version	Date	Author	Changes
0.1	2026-02-23	John (AI Director)	Initial draft — Bilko change control process

---

> **NOTE:** This file documents the **Bilko Change Request Process** and provides the template for individual Change Requests. Copy the CR Template section below for each new CR, replacing all fields. Save individual CRs as `CR-XXX-title.md` in the same directory.

Bilko Change Control Overview

Bilko is an accounting SaaS with regulatory compliance requirements (SEF, PDV, GDPR). Changes to approved requirements carry additional risk because:

1. **Accounting law compliance** — a scope change to invoice or VAT logic may break Serbian legal requirements
2. **Double-entry integrity** — changes to the transaction model can corrupt financial records
3. **SEF API compatibility** — invoice format changes may cause SEF rejections for existing users

Therefore: ALL scope, requirement, budget, or timeline changes must go through this process. No exceptions for "small" changes.

Change Request Log

CR ID	Title	Submitted By	Date	Type	Impact	Status	Decision Date
CR-001	—	—	—	—	—	—	—

> Add each CR to this log when submitted. Keep this file as the single-page overview; individual CR details in separate files.

Approval Matrix

Budget / Timeline Impact	Required Approvals	Target Decision Time
None	John	1 business day
< 5% budget OR < 3 days timeline	John	1 business day
5–15% budget OR 3–14 days	John → Alem	3 business days
> 15% budget OR > 14 days	John + Alem	5 business days
Regulatory change (SEF, PDV, GDPR)	John + Alem + Asmir	5 business days

CR Template — Copy Below for Each New Change Request

Change Request: [CR Title — short imperative description e.g. "Add Croatia PDV 25% rate

support"]

> **Project:** Bilko — Balkan Accounting SaaS **CR ID:** CR-[NNN] *(assigned by John — use next sequential number from the CR log)* **Version:** 0.1 **Date:** 2026-02-23 **Author:** John (AI Director) **Status:** Draft **Reviewers:** Alem Bašić (CEO)

1. Change Request Metadata

Field	Value
CR ID	CR-[NNN]
Date Submitted	2026-02-23
Submitted By	John (AI Director)
Project	Bilko
Priority	Critical / High / Medium / Low
CR Type	Scope Change / Budget Change / Timeline Change / Requirements Change / Regulatory Change / Technical Change
Current Phase	Planning / Design / Development / Testing / Deployment
Decision Deadline	[DATE — by when Alem must decide to avoid delaying the next sprint or launch milestone]

---

2. Change Description

2.1 Summary of Change

[One paragraph: what is changing, why it is changing, and what the expected outcome is. Be specific — name the feature, module, or regulation driving the change. Example: "The Serbian Tax Administration updated the SEF e-invoice schema (version 2.1) effective 2026-04-01. The current Bilko SEF integration targets schema version 2.0. Without this change, all invoice submissions will be rejected by SEF from April 1, resulting in legal non-compliance for all Serbian Bilko users."]

2.2 Current State (Before)

[Describe the current approved state in concrete terms. What does the system do today? What requirement, architecture, or document describes this current behavior? Example: "The invoice submission module (apps/api/src/routes/invoices/) generates SEF XML using schema v2.0. The UBL namespace is 'urn:oasis:names:specification:ubl:schema:xsd:Invoice-2' as specified in FR-042."]

Currently approved in: `[Document name and version e.g. functional-requirements.md v1.2]`, Section [X.X]

2.3 Proposed State (After)

[Describe what the system should do after this change, with enough detail for implementation. Include field names, schema names, API changes, or UI changes as appropriate. Example: "Update the SEF XML generator to produce schema v2.1. New required field: cbc:ProfileID urn:fdc:sef.gov.rs:2021:billing</cbc:ProfileID>. Update the validator to check v2.1 compliance before submission. No UI changes required — the change is in the API layer only."]

2.4 Out of Scope for This Change

This CR does NOT include:

- [First explicitly excluded item — something adjacent that might be assumed but is not in scope. Example: "BiH or Croatia e-invoice format changes — these are tracked separately."]
- [Second explicitly excluded item. Example: "UI changes to the invoice creation wizard — no user-visible changes in this CR."]

---

3. Reason & Justification

3.1 Reason for Change

Reason Category	Applies?	Details
New business requirement discovered	Yes / No	[Describe if applicable, otherwise "N/A"]
Regulatory / compliance mandate (SEF, PDV, GDPR)	Yes / No	[Which regulation, effective date, enforcement body — e.g. "SEF schema v2.1 mandatory from 2026-04-01 per RS Tax Administration notice"]
Technical blocker / infeasibility	Yes / No	[What technical limitation was discovered and why it blocks progress]

Reason Category	Applies?	Details
Market opportunity / competitive pressure	Yes / No	[Market context if applicable]
Error/omission in original requirements	Yes / No	[What was missed in the original spec and when discovered]
Beta user feedback	Yes / No	[Which beta user(s) raised the issue, what they reported]

Primary Justification: [One clear statement of why this change must happen. Focus on business impact: revenue risk, legal obligation, user blocking issue, or architectural necessity. Example: "Serbian law requires SEF-compliant invoices. Schema v2.0 becomes invalid on 2026-04-01. All Serbian users will face fines and invoice rejections if Bilko does not update. This is a non-negotiable legal compliance change."]

3.2 Consequence of Not Changing

If this change is **not** approved: [Concrete consequences — be specific about legal, financial, or user impact. Example: "All Serbian Bilko users will be unable to submit e-invoices to SEF from April 1. This exposes users to fines under Serbian tax law and exposes ALAI to liability. Bilko becomes non-compliant and unsellable in the Serbian market."]

---

4. Impact Analysis

4.1 Scope Impact

Deliverable Affected	Current Scope	Proposed Scope	Impact Type
[Name of module, feature, or document affected e.g. "SEF XML generator"]	[What it does today]	[What it will do after the change]	Added / Removed / Modified

Scope Change Size: Small (< 1 day) / Medium (1–3 days) / Large (3–10 days) / Major (> 10 days)

4.2 Timeline Impact

Milestone	Current Date	New Date (if approved)	Delay
[Milestone name affected by this CR e.g. "Backend: SEF integration complete"]	[Current planned date]	[New date if CR approved]	[N] days
Serbia production launch	2026-05-01	[New date if delayed, or "No change"]	[N] days

Timeline Impact: None / Minor (≤ 3 days) / Moderate (4–14 days) / Major (> 14 days) **Critical Path Impact:** Yes / No

4.3 Budget Impact

Cost Category	Current Budget (EUR)	Additional Cost (EUR)	Notes
Development	[Current allocated amount]	[Extra hours × rate, or 0 if within capacity]	[Explain if using builder agents or external dev]
Testing	[Current allocated amount]	[Extra testing effort cost, or 0]	
Total Additional Cost		**[Sum of additional costs]**	

Budget Impact: None / Minor (< 5%) / Moderate (5–15%) / Major (> 15%)

4.4 Accounting/Regulatory Impact

Area	Impact	Details
SEF compatibility	Affected / Not Affected	[If affected: which SEF fields or schema version changes; if not: "No changes to invoice submission flow"]
PDV calculation logic	Affected / Not Affected	[If affected: which VAT rates or calculation rules change; if not: "No changes to PDV computation"]
Double-entry integrity	Affected / Not Affected	[If affected: which debit/credit posting rules change; must be verified with accounting test dataset]
Audit trail (LoggedAction)	Affected / Not Affected	[If affected: which new actions need to be logged; LoggedAction is append-only]
GDPR compliance	Affected / Not Affected	[If affected: which personal data fields are added, removed, or accessed differently]
Balkan GAAP	Affected / Not Affected	[If affected: which accounting standard rules (RS/BA/HR) this touches]

Regulatory review required: Yes / No **If yes, reviewer:** Asmir Merdžanović (SnowIT) + Alem

4.5 Risk Impact

Risk	Probability	Impact	Notes

[New risk introduced by this change e.g. "SEF migration introduces regression risk on existing invoices"]	H/M/L	H/M/L	Add to risk-register.md
[Existing risk affected by this change e.g. "SEF rejection risk — currently Medium, may change"]	H/M/L	H/M/L	Update risk-register.md entry

4.6 Quality Impact

- Test cases affected: [List TC IDs needing update e.g. "TC-042 (SEF XML validation), TC-043 (invoice submission end-to-end)"]
- Regression risk: High / Medium / Low — [Explain why e.g. "Medium — changes XML generation layer only; no changes to PDV logic or double-entry posting"]
- NFRs affected: [List any non-functional requirements impacted e.g. "NFR-07 (SEF submission < 2s) — must re-test performance with new schema"]
- Double-entry balance check: Affected / Not Affected

4.7 Affected Deliverables / Documents

Document	Section	Type of Change	Owner
`functional-requirements.md`	FR-[NNN — the FR this CR modifies]	Modify	John
`user-stories.md`	US-[NNN — affected user stories]	Add/Modify	John
`acceptance-criteria.md`	AC-[NNN — ACs that need updating]	Update	John
`requirements-traceability-matrix.md`	Row for FR-[NNN]	Update	John
`risk-register.md`	R-[NNN — risk entries to add/update]	Add/Update	John
`PIPELINE.md`	Status	Update	John
`packages/database/prisma/schema.prisma`	Model [ModelName — if schema change required]	Modify	Tech Lead
Test cases	TC-[NNN — test cases to update or add]	Update / Add	QA

---

5. Alternative Approaches Considered

Alternative	Description	Why Rejected
Option A (Proposed)	[Describe this CR's proposed solution]	*Recommended*
Option B	[Describe an alternative approach considered e.g. "Delay SEF update to post-launch, notify users manually"]	[Why this alternative is inferior e.g. "Does not meet legal deadline; exposes users to fines"]
Option C — Do Nothing	Reject the change; proceed as originally planned	[State the consequence e.g. "Serbian market becomes non-compliant; all SEF submissions fail from April 1"]

Recommendation: Option A / B / C **Rationale:** [One sentence on why the recommended option is the best choice given cost, risk, timeline, and compliance tradeoffs]

---

6. Implementation Plan

6.1 Implementation Steps

#	Task	Owner	Effort	Target Date
1	[Primary implementation task — describe the code change e.g. "Update SEF XML generator to produce schema v2.1"]	John / builder agent	[Estimated effort e.g. "4h"]	[Target date]
2	Update affected FR/US/AC documents	John	1h	[Target date]
3	Update test cases	validator agent	[Estimated effort]	[Target date]
4	If schema change: create Prisma migration	builder agent	[Estimated effort, or "N/A if no schema change"]	[Target date]
5	Regression testing	validator agent	[Estimated effort]	[Target date]

6.2 Dependencies

Dependency	Type	Blocking?
[Name any dependency this CR has e.g. "SEF schema v2.1 spec published by RS Tax Administration" or "Prisma migration must run before deployment"]	Internal / External / Regulatory	Yes / No

6.3 Test Plan for This Change

- Unit tests: [List which unit test files need new or updated tests e.g. "sef-xml-generator.test.ts — add tests for v2.1 fields"]
- Integration tests: [List which integration tests to re-run e.g. "Invoice submission end-to-end against SEF sandbox"]
- Accounting validation: [If financial logic changed: describe the test dataset and balance verification. If not: "Not required — no changes to PDV or double-entry logic"]
- SEF sandbox test: [If SEF affected: "Yes — must pass full submission cycle in SEF test environment before production deploy". If not: "Not required"]
- Regression scope: [Which existing features to regression-test e.g. "All invoice creation paths, PDF generation, existing SEF submissions"]

---

7. Rollback Plan

Rollback Trigger: [Define the condition that would trigger a rollback e.g. "SEF rejection rate exceeds 5% within first 24h of production deploy" or "Balance sheet imbalance detected in accounting integrity check"]

Rollback Steps:

1. Revert to previous deployment (git revert or container rollback)
2. Run accounting integrity check: `SELECT SUM(debit) - SUM(credit) FROM transactions` — must be 0
3. Notify affected users if data was migrated
4. Restore DB to pre-migration snapshot if schema was changed

Rollback Owner: John **Rollback Time Required:** < 1 hour for code rollback; up to 4 hours if DB migration involved **Data Recovery Needed:** Yes (if schema migration) / No (if code-only)

---

8. Approval Workflow

8.1 This Change Requires

☐ **John Review** — Impact analysis complete and accurate

☐ **Regulatory Review (Asmir)** — Only if SEF, PDV, or accounting law affected

☐ **Alem (CEO)** — Budget > 15% OR timeline > 14 days OR regulatory decision

8.2 Decision Record

Level	Reviewer	Decision	Date	Comments
AI Director (John)	John	Approved / Rejected / Deferred	[Date of decision]	[Any conditions, concerns, or notes]
Regulatory (Asmir)	Asmir Merdžanović	Approved / Rejected	[Date of decision]	*(if required — only for SEF/PDV/GDPR changes)*
CEO (Alem)	Alem Bašić	Approved / Rejected	[Date of decision]	*(if required — budget > 15% or timeline > 14 days)*

Final Decision: APPROVED / REJECTED / DEFERRED **Decision Date:** [Date the final decision was recorded] **Effective From Sprint:** Sprint [Sprint number when implementation begins]

---

9. Change Log

Date	Changed By	What Changed
2026-02-23	John	Template initialized

Lessons Learned

Lessons Learned: Bilko

> **Project:** Bilko — Balkan Accounting SaaS **Version:** 0.1 **Date:** 2026-02-23
> **Author:** John (AI Director) **Status:** Draft **Reviewers:** Alem Bašić (CEO)

Document History

Version	Date	Author	Changes
0.1	2026-02-23	John (AI Director)	Initial entry — Phase 1 kickoff learnings

---

> "Lessons learned without behavior change is not learning." — Alem Bašić, 2026-02-13 (ZAKON #0)

Every entry in this log must produce either a systemic fix (hook/tool/rule) or a concrete behavior change. Observations without action items do not belong here.

---

1. Session Metadata — Phase 1 Kickoff Review

Field	Value
Session Type	Phase Review (after Pipeline Gate 8 pass + initial build)
Project	Bilko
Phase / Sprint	Pre-Sprint 1 — Phase 1 kickoff
Period Covered	2026-02-19 – 2026-02-23
Facilitator	John (AI Director)

Field	Value
Participants	John, Alem Bašić
Date of Session	2026-02-23
Duration	Retrospective of first 4 days

2. What Went Well (Keep Doing)

#	What Went Well	Why It Worked	How to Preserve	Category
KD-01	8-gate pipeline forced all critical research before coding began	Gates prevented building on wrong assumptions; regulatory research (MULTI-REGION-OVERVIEW.md) was detailed and accurate	Continue using pipeline gates for Phase 2 (Croatia) and Phase 3 (BiH); enforce gate passage before any new country's features	Process
KD-02	Database schema (15 models) designed before writing any application code	Schema aligned with accounting law requirements (double-entry, NUMERIC, audit trail) from day 1; no retroactive migrations	Always design data model first for financial features; NEVER write API code against an undefined schema	Technical
KD-03	NUMERIC(19,4) for all monetary fields decided upfront and enforced in schema	Prevented entire class of floating-point accounting errors; found in Fiken research	CI test: fail if any new field for monetary values uses Float/Decimal without explicit 4dp precision; document in CLAUDE.md	Technical
KD-04	Regulatory research completed and documented before building	SEF API requirements, PDV rates, Croatian eRačun timing — all known before design phase	Research-first mandate in project charter; Asmir as ongoing regulatory advisor	Process
KD-05	CEO approval gate (Gate 8) prevents building orphaned features	Alem reviewed full plan before code was written; GO decision is documented and traceable	Maintain Gate 8 for each major phase/country expansion	Process

Highlights

KD-02 Detail: The decision to define all 15 Prisma models (Organization, User, Invoice, InvoiceItem, Transaction, etc.) before writing a single API endpoint proved to be the most valuable architectural decision of Phase 1. When the backend starts being built, the developer (human or AI) has a complete, validated schema to work against — no guessing about data shapes. The schema covers double-entry (debitAccountId + creditAccountId in Transaction), multi-currency (ExchangeRate model with locked rates), and audit trail (LoggedAction append-only). This pattern should be standard for all future Bilko features.

---

3. What Didn't Go Well (Stop Doing)

#	What Didn't Work	Root Cause	Impact (1-10)	Action Item	Owner	Deadline
SD-01	Frontend built entirely with mock data — 8 pages, zero real API connections	Prototype-speed MVP prioritized demo-ability over production-readiness; no explicit decision to replace mock data before launch	10	CI check: grep for mock-data.ts imports outside test files; fail build in staging/production	John	Sprint 1
SD-02	packages/ui component library left empty — all components in apps/web, creating future duplication risk	"We'll extract later" reasoning; never happened during MVP build	4	Document as TD-003; schedule for Phase 2 before mobile work starts	John	2026-05-01
SD-03	No CI/CD pipeline established before writing application code	Infrastructure "can wait" assumption; now blocking safe iteration	8	CI/CD setup is Sprint 1 task 1 before any API endpoints	John	2026-03-07

#	What Didn't Work	Root Cause	Impact (1-10)	Action Item	Owner	Deadline
SD-04	API-REFERENCE.md not created alongside the schema and routes structure	Documentation always deferred in early phases; creates frontend/backend misalignment	5	Every new API endpoint must have its Request/Response schema documented in API-REFERENCE.md in the same PR	John	Ongoing
SD-05	No unit tests written during MVP build	Speed prioritized; tests "added later" — never happened	9	Test must be written in the same sprint as the feature, not deferred	John	Ongoing

Root Cause Analysis — Deep Dives

SD-01: Mock Data Throughout Frontend

Problem: All 8 frontend pages show hardcoded mock data. Beta testers would see fake numbers. Cannot test real API integration.

5 Whys:

1. Why is all data mocked? → Frontend was built as a prototype to show design/flow, not as production code
2. Why was it built as prototype only? → Gate 8 (CEO approval) was needed before real backend investment; prototype reduced risk
3. Why wasn't mock replacement scheduled explicitly? → The PIPELINE.md "Next Steps" listed it but no MC task was created with a deadline
4. Why was no MC task created? → Prototype phase didn't have sprint planning; items added to informal list
5. Why did informal list not get actioned? → No hook or gate enforcing the replacement before beta access

Root Cause: No automated enforcement that mock data must be replaced before staging deployment. **Systemic Fix:** CI check in `apps/api/src/` and `apps/web/` — grep for `mock-data.ts` imports outside `/test/` directories; fail build if found in staging/production environment. Add to `.github/workflows/ci.yml`. **Ljestvica Fikseva (ZAKON #1):** Hook > Tool > Rule > CLAUDE.md > Memory → Using CI hook (strongest enforcement)

SD-05: No Tests Written During MVP Build

Problem: 0% test coverage on financial logic (PDV, double-entry, SEF). Accounting software without tests is a liability — a PDV calculation bug could cause users to file incorrect tax returns.

5 Whys:

1. Why are there no tests? → They were "deferred" to after the feature was "working"
2. Why were they deferred? → Speed pressure to show CEO a working prototype
3. Why did "working" not include tests? → No definition of "done" included test coverage
4. Why was there no DoD? → Project was in prototype phase with informal standards
5. Why were informal standards accepted? → No enforcement mechanism (no CI, no PR gate)

Root Cause: No CI gate requiring tests before merge; no DoD including test coverage. **Systemic Fix:** GitHub Actions CI: `npm test -- --coverage --coverageThreshold='{"global":{"lines":80}}'` — fails PR if coverage below 80% overall or below 95% for `src/services/` (financial logic). This is non-negotiable before Sprint 2 merges. **Ljestvica Fikseva:** CI coverage gate → fails PR merge.

---

4. What to Try Next (Start Doing)

#	What to Try	Expected Benefit	Hypothesis	Success Metric	Trial Period
ST-01	Write tests FIRST (TDD) for financial logic (PDV, double-entry, SEF)	Catches accounting bugs before they reach users; forces clear spec before coding	If we write Given/When/Then tests for PDV calculation before writing the calculation, then PDV accuracy errors drop to 0	Zero PDV calculation bugs in beta; CI balance check passes 100%	Sprint 2-4
ST-02	Create API-REFERENCE.md endpoint-by-endpoint alongside backend implementation	Eliminates frontend/backend sync issues; enables frontend developer (or agent) to start immediately	If every endpoint has documented Request/Response before frontend implements it, then integration time drops 50%	Zero "unexpected API shape" issues during frontend-backend connection sprint	Sprint 2
ST-03	SEF sandbox test on every PR touching invoice logic	Catches SEF breaking changes before they reach production users	If we run SEF sandbox test in CI for every invoice change, then SEF regression bugs = 0 in production	Zero SEF regressions post-launch	Sprint 3 (when SEF built)

---

5. Key Findings by Category

5.1 Technical Findings

#	Finding	Impact	Recommendation
TF-01	Prisma's Decimal type maps cleanly to NUMERIC(19,4) — zero extra work for financial precision	High positive	Continue using Decimal for all monetary Prisma fields; never use Float
TF-02	LoggedAction as append-only audit table is a strong pattern — enforces immutability at schema level	High positive	Apply same pattern to other sensitive tables in future products (Drop, BasicFakta)
TF-03	Turborepo monorepo structure enables shared packages (database, ui) but requires discipline — empty packages create confusion	Medium	Either fill packages immediately or don't create them; no empty scaffolds
TF-04	Next.js 15 App Router + React Server Components significantly changes how data fetching works — team must understand before building	High	Document RSC vs client component decision boundary in CLAUDE.md

Technical Patterns Observed:

- "Schema first, endpoints second, frontend third" — the correct build order for financial software; schema ensures all features are feasible
- Monorepo shared packages need a charter: what goes in packages/ui, what stays in apps/web — decide before Sprint 2

5.2 Process Findings

#	Finding	Root Cause	Recommendation
PF-01	8-gate pipeline prevented scope creep effectively — every decision was documented before coding	Explicit gate criteria with documented evidence	Apply same pipeline to Croatia (Phase 2) and BiH (Phase 3); don't skip gates even if process feels familiar

#	Finding	Root Cause	Recommendation
PF-02	"Next Steps" in PIPELINE.md without MC tasks = invisible work	No translation from documentation to task management	Every "Next Step" in any doc must have a corresponding MC task created immediately
PF-03	Prototype → production transition requires explicit decision and enforcement, not assumption	Prototype mode is sticky — it continues until something forces a change	Launch gate checklist must include "zero mock data imports verified by CI"

Process Patterns Observed:

- Documentation-first works for planning but requires enforcement mechanisms to drive action
- Sprint planning discipline is critical even for AI-only teams — without sprints, work becomes unordered

5.3 Communication Findings

#	Finding	Stakeholder Impact	Recommendation
CF-01	Weekly Slack summary to Alem keeps CEO aligned without overhead meetings	Alem can make informed go/no-go decisions asynchronously	Maintain weekly sprint summary format from Communication Plan
CF-02	Regulatory questions need Asmir's input — cannot be resolved by research alone	Wrong SEF implementation would cause launch failure	Asmir review at every SEF-touching sprint; not just at start

5.4 Tools & Infrastructure Findings

#	Finding	Impact	Recommendation
IF-01	No CI/CD pipeline means every deploy is a manual risk	High — production incidents have no automated safety net	CI/CD is Sprint 1 task 1, not an afterthought
IF-02	SEF sandbox environment must be set up before SEF integration sprint begins	Medium — delay if sandbox access not ready	Alem/Asmir to secure SEF sandbox credentials by 2026-03-01

5.5 Team & Collaboration Findings

#	Finding	Impact	Recommendation

TM-01	AI agent team (John + builder/validator agents) works well for structured implementation tasks but requires extremely detailed specs — vague requirements produce vague code	High	Every FR must have explicit Acceptance Criteria before being assigned to a builder agent
TM-02	Financial domain requires accounting expertise that pure coding agents don't have — PDV, double-entry, SEF need validation by someone who knows the law	Critical	Asmir review for all regulatory features; automated accounting logic tests are the safety net

---

6. Action Items

#	Action Item	Category	Owner	Sprint / Deadline	Priority	Status
AI-01	Set up CI/CD pipeline (GitHub Actions) — include: lint, type-check, tests, coverage gate, mock-data import check	Infrastructure	John	Sprint 1 (by 2026-03-07)	P1	Open
AI-02	Write unit tests for tax.service.ts (PDV calculation) as first test file — TDD model	Testing	builder agent	Sprint 2 start (2026-03-07)	P1	Open
AI-03	Create `apps/api/docs/API-REFERENCE.md` — document every endpoint as it is built	Documentation	John	Ongoing from Sprint 1	P2	Open

#	Action Item	Category	Owner	Sprint / Deadline	Priority	Status
AI-04	Add CI check: grep for `mock-data.ts` in non-test files — fail build if found in staging env	Tools/Infrastructure	John	Sprint 1 (by 2026-03-07)	P1	Open
AI-05	Secure SEF sandbox credentials from APR via Asmir	Regulatory	Alem + Asmir	By 2026-03-01	P1	Open
AI-06	Create MC tasks for every "Next Step" in PIPELINE.md	Process	John	Now	P2	Open
AI-07	Asmir to review SEF integration design before Sprint 3 implementation	Regulatory	John	By 2026-03-14	P1	Open
AI-08	Define RSC vs client component decision boundary for Bilko in CLAUDE.md	Technical	John	Sprint 1	P2	Open

---

7. Metrics Comparison — Planned vs. Actual

7.1 Timeline (Phase 1 kickoff only)

Milestone	Planned	Actual	Variance	Root Cause
All 8 pipeline gates	2026-02-19	2026-02-20	+1 day	Gate 5-8 validation took longer than expected; thorough — good

Milestone	Planned	Actual	Variance	Root Cause
Backend foundation (auth, 4 endpoints)	2026-02-20	2026-02-20	0	On target
Full 50 endpoints	2026-03-07	TBD	TBD	Sprint 1

7.2 Budget (Phase 1 so far)

Category	Planned (EUR)	Actual (EUR)	Variance	Notes
Research + design + schema	~€1,500	~€800	-€700	AI-assisted research more efficient than estimated
Backend foundation (auth)	~€500	~€300	-€200	4 endpoints in 1 session
Running total	**~€2,000**	**~€1,100**	**-€900**	Under budget so far

7.3 Quality

Metric	Target	Actual	Status
Test coverage	≥ 80%	0% (not yet written)	In progress
Critical bugs at this stage	0	0	
Pipeline gate pass rate	8/8	8/8	
Database schema completeness	15 models	15 models	

---

8. Recommendations for Future Projects

#	Recommendation	Category	Confidence	Applicable When
REC-01	Define all database models before writing any API endpoints or frontend code	Technical	High	Any data-driven application; especially financial
REC-02	Use pipeline gates (8-gate or adapted) before investing in build phase for any new product	Process	High	All new ALAI products

#	Recommendation	Category	Confidence	Applicable When
REC-03	Set up CI/CD in Day 1 of Sprint 1 — never defer infrastructure; it becomes load-bearing	Technical	High	Every project
REC-04	For regulated domains (accounting, finance, health): TDD is not optional; tests are the compliance proof	Technical	High	Any compliance-regulated feature
REC-05	Research regulatory requirements with a local expert (Asmir for Balkans) before designing features that touch them	Process	High	Any Balkan market expansion

Top 3 Most Important Learnings

1. **Schema First, Always:** The 15-model Prisma schema designed before writing a single line of application code is why Bilko's data architecture is sound. The schema was validated against accounting law (double-entry), PRD features, and multi-currency requirements all at once. Future features should follow the same discipline: design the data model, get it reviewed, then build.
2. **Mock Data Is a Prototype Trap:** Shipping a frontend with mock data creates a false sense of completeness. The 8 pages look done but nothing works end-to-end. In future builds: either build frontend and backend in parallel (stub real API from day 1), or explicitly schedule "replace mock with real" as a first-class sprint task with CI enforcement.
3. **Regulatory Compliance Is Not a Sprint Task — It's an Ongoing Advisor Relationship:** SEF is not just "call an API." It's a living government system with evolving requirements, certificate requirements, and edge cases that only practitioners know. Asmir's involvement is not optional — it's risk mitigation. Budget time for regulatory review at every sprint touching financial compliance.

---

9. Knowledge Base Contribution

Finding	Contribution Type	Target System	Status	Owner
NUMERIC(19,4) for all monetary fields	Rule added to Bilko CLAUDE.md	`~/ALAI/products/Bilko/CLAUDE.md`	Done (already in CLAUDE.md)	John

Finding	Contribution Type	Target System	Status	Owner
Pipeline gate process	Template	`~/ALAI/products/Bilko/PIPELINE.md`	Done	John
Mock data replacement pattern	HiveMind entry	HiveMind	Open	John
Schema-first pattern	HiveMind entry	HiveMind	Open	John

HiveMind entries to add:

```
node ~/system/agents/hivemind/hivemind.js post "Bilko pattern: Always design Prisma schema before API endpoints. Schema = contract. Validated against accounting law + PRD features before writing any code."
node ~/system/agents/hivemind/hivemind.js post "Bilko lesson: Mock data in frontend is a prototype trap. Enforce removal with CI grep check before staging deployment. See CROSS-CUTTING/tech-debt-log.md TD-001."
```

Approval

Role	Name	Date	Signature
Author	John (AI Director)	2026-02-23	
Reviewer			
Project Manager	John	2026-02-23	
AI Director (John)	John	2026-02-23	
Team	Alem Bašić		

Tech Debt Log

Tech Debt Log: Bilko

> **Project:** Bilko — Balkan Accounting SaaS **Version:** 0.1 **Date:** 2026-02-23
> **Author:** John (AI Director) **Status:** Draft **Reviewers:** Alem Bašić (CEO)

Document History

Version	Date	Author	Changes
0.1	2026-02-23	John (AI Director)	Initial population — known debt from Phase 1 kickoff

---

1. Tech Debt Philosophy

Technical debt is not inherently bad. The Bilko MVP was built quickly (by design) to validate market fit before spending full engineering resources. The known debts below were **conscious decisions** — the correct ones at the time. This log exists to make them visible, prioritizable, and tracked to resolution.

Bilko-specific debt policy:

- **Financial logic debt (double-entry, PDV, SEF):** Zero tolerance — fix immediately. Accounting errors have legal consequences for users.
- **UI/UX debt (mock data, placeholder pages):** High priority — block beta launch if not resolved.
- **Infrastructure debt (CI/CD, monitoring):** High priority — block production launch if not resolved.
- **Documentation debt:** Medium — resolve before Croatia launch.

---

2. Summary Dashboard

Metric	Current	Last Sprint	Trend
Total Items	11	0	↑ (initial population)
Open	11	0	↑
In Progress	0	0	→
Resolved	0	0	→
Critical / High (P1/P2)	5	0	↑
Debt in current sprint	0	0	→
Avg age of open items (days)	3 (project just started)	—	—
Debt resolved this sprint	0	0	→

Sprint Debt Budget: 20% of sprint capacity allocated to debt reduction starting Sprint 2.

---

3. Tech Debt Categories

Category	Description	Common Causes	Detection Method
Design	Wrong abstractions, poor separation	Speed, evolving requirements	Code review
Code Quality	Duplication, complexity, poor naming	Speed over quality	Static analysis
Infrastructure	Manual processes, missing monitoring	Growth outpacing ops maturity	Ops review
Documentation	Missing or outdated docs	Documentation deferred	New team feedback
Dependency	Outdated libraries, deprecated APIs	Insufficient update cadence	Snyk/Dependabot
Testing	Low coverage, missing tests, flaky tests	Speed over quality	Coverage reports
Security	Known vulnerabilities, security shortcuts	Urgency	Security scans
Performance	N+1 queries, no caching, bottlenecks	Optimizations deferred	Performance monitoring

---

4. Impact Assessment Methodology

4.1 Impact Score (1–10)

Score	Impact Level	Effect on Team/Business
1–2	Negligible	Cosmetic; no measurable effect
3–4	Minor	Slightly slows specific tasks; <5% velocity impact
5–6	Moderate	Slows multiple tasks; 5–15% velocity impact
7–8	High	Significantly slows development; recurring bugs; 15–30% impact
9–10	Critical	Blocks progress or creates legal/financial risk for users

4.2 Effort to Fix (Story Points)

Size	Points	Typical Scope
XS	1	Single line / config change
S	2–3	Single function / component refactor
M	5–8	Module-level changes
L	13	Multi-module refactoring
XL	21+	Major refactor; break into sub-tasks first

4.3 Priority Calculation

Priority	Criteria	Sprint Allocation Target
P1 — Critical	Impact ≥ 8 OR blocks beta/launch	Schedule within 1 sprint
P2 — High	Impact 6–7 OR blocks a planned feature	Schedule within 2 sprints
P3 — Medium	Impact 4–5	Schedule within 4 sprints
P4 — Low	Impact ≤ 3	Schedule when convenient

---

5. Tech Debt Register

ID	Description	Category	Introduced	Impact (1-10)	Effort	Priority	Owner	Status	Target Resolution	Notes
TD-001	**ALL frontend data served from mock-data.ts** — 8 pages (Dashboard, Invoices, Expenses, Purchases, Banking, Reports, VAT, Settings) use hardcoded mock data instead of real API calls	Code Quality	2026-02-19 (MVP build)	10	L (13)	P1	John	Open	2026-03-14	BLOCKS beta launch. CI must grep for mock-data.ts imports and fail build in staging/production

ID	Description	Category	Introduced	Impact (1-10)	Effort	Priority	Owner	Status	Target Resolution	Notes
TD-002	**Backend API 0/50 core endpoints** — Only 4 auth endpoints built. Invoice, expense, contact, account, transaction, report, banking endpoints all missing	Design	2026-02-20 (MVP build)	10	XL (50 endpoints)	P1	John	In Progress	2026-03-14	Primary Sprint 1-2 work. 4/50 done (auth).
TD-003	**packages/ui empty scaffold** — Shared UI component library initialized but empty. All components live inside apps/web directly, causing potential duplication if mobile app added	Design	2026-02-19	4	M (8)	P3	John	Open	2026-05-01	Not urgent for Phase 1. Becomes critical if React Native added in Phase 2

ID	Description	Category	Introduced	Impact (1-10)	Effort	Priority	Owner	Status	Target Resolution	Notes
TD-004	**No unit tests written** — 0% test coverage on the entire codebase. Double-entry logic, PDV calculation, and SEF integration have zero automated verification	Testing	2026-02-20	9	L (13 for critical paths)	P1	John	Open	2026-03-07	Financial software without tests is a legal risk. CI must enforce ≥80% coverage before merge to main
TD-005	**No CI/CD pipeline** — No GitHub Actions or equivalent. All deployment is manual. No automated test runs on PR. No staging environment defined	Infrastructure	2026-02-20	8	M (8)	P1	John	Open	2026-03-07	Blocks safe iteration. Must be operational before any team member merges code

ID	Description	Category	Introduced	Impact (1-10)	Effort	Priority	Owner	Status	Target Resolution	Notes
TD-006	**No monitoring or alerting** — No uptime monitor, no error tracking (Sentry), no APM tool configured. SEF submission failures would go undetected	Infrastructure	2026-02-20	8	S (3)	P2	John	Open	2026-03-21	Minimum: uptime monitor + Sentry for production launch
TD-007	**SEF integration not built** — SefService placeholder exists conceptually but no actual SEF API integration implemented. No UBL 2.1 XML generation	Design	2026-02-20	10	L (13)	P1	John	Open	2026-03-21	Serbia launch blocker. Must be fully tested in SEF sandbox before launch

ID	Description	Category	Introduced	Impact (1-10)	Effort	Priority	Owner	Status	Target Resolution	Notes
TD-008	**No database indexes on high-frequency queries** — Prisma schema has no explicit indexes. Invoice list query filtering by organizationId + status will be slow at scale (>10K records)	Performance	2026-02-20	6	S (3)	P2	John	Open	2026-03-28	Add indexes on: Invoice(organizationId, status), Transaction(organizationId, date), Expense(organizationId)

ID	Description	Category	Introduced	Impact (1-10)	Effort	Priority	Owner	Status	Target Resolution	Notes
TD-009	**No environment configuration validation** — Application starts even if critical env vars missing (DATABASE_URL, JWT_SECRET, SEF_API_KEY). Silent failures in production	Infrastructure	2026-02-20	7	XS (1)	P2	John	Open	2026-03-07	Add startup validation: fail fast if required env vars absent
TD-010	**i18n not implemented** — All text hardcoded in English in frontend components. Serbian language support (core requirement BR-011) not yet implemented	Code Quality	2026-02-19	7	M (8)	P2	John	Open	2026-03-28	Need next-intl or similar. Serbian (Latin) required for beta. Cyrillic optional initially

ID	Description	Category	Introduced	Impact (1-10)	Effort	Priority	Owner	Status	Target Resolution	Notes
TD-011	**API-REFERENCE.md not created** — Backend API contract documentation referenced in CLAUDE.md does not exist yet. API endpoints have no documented request/response schemas	Documentation	2026-02-20	5	M (5)	P3	John	Open	2026-03-14	Should be created alongside API endpoints. Prevents frontend/backend sync issues

Status Values: Open | In Progress | Resolved | Accepted (deferred) | Won't Fix (justified)

---

6. Prioritization Framework

6.1 Cost of Delay Analysis

ID	Description	Cost Per Sprint (velocity loss %)	Sprints Open	Accumulated Risk
TD-001	Mock data in frontend	30% — all features untestable with real data	1	HIGH — beta shows fake data
TD-004	No unit tests	15% — regressions catch later, cost 3× to fix	1	HIGH — accounting bugs undetected

ID	Description	Cost Per Sprint (velocity loss %)	Sprints Open	Accumulated Risk
TD-005	No CI/CD	20% — manual deploy risk of breaking production	1	CRITICAL — no safety net
TD-007	SEF not built	100% (Serbia launch impossible)	1	CRITICAL — launch blocker

6.2 Debt Payoff Priority Order

1. **TD-002** (Backend endpoints) — Sprint 1-2 focus, primary sprint work
2. **TD-007** (SEF integration) — Sprint 2-3, Serbia launch blocker
3. **TD-001** (Mock data replacement) — Sprint 2-3, coincides with TD-002
4. **TD-004** (Unit tests) — Ongoing from Sprint 1; written alongside features
5. **TD-005** (CI/CD) — Sprint 1, early; safety net for all subsequent work
6. **TD-009** (Env validation) — XS, quick win Sprint 1
7. **TD-008** (DB indexes) — S, pre-launch
8. **TD-010** (i18n) — Sprint 2-3, beta blocker
9. **TD-006** (Monitoring) — Sprint 3, pre-launch
10. **TD-011** (API docs) — Ongoing, Sprint 2
11. **TD-003** (Shared UI package) — Deferred to Phase 2

---

7. Sprint Debt Allocation Guidelines

Sprint	Debt Items to Address	Rationale
Sprint 1 (Feb 23 – Mar 7)	TD-005 (CI/CD), TD-009 (env validation), TD-004 (test framework setup)	Safety net before building features
Sprint 2 (Mar 7-14)	TD-001 (mock → real API), TD-002 (backend endpoints), TD-010 (i18n scaffold)	Primary feature work + debt concurrent
Sprint 3 (Mar 14-21)	TD-007 (SEF), TD-008 (DB indexes)	Serbia launch readiness
Sprint 4 (Mar 21-28)	TD-006 (monitoring), TD-011 (API docs)	Production quality
Sprint 5 (Alpha/Beta)	TD-004 (remaining test coverage to ≥80%)	Pre-launch quality gate

---

8. Debt Reduction Tracking

Sprint	Debt Items Added	Debt Items Resolved	Net Change	Running Total Open
Sprint 0 (kickoff)	11	0	+11	11
Sprint 1	TBD	TBD	TBD	TBD
Sprint 2	TBD	TBD	TBD	TBD

Debt Trend: INCREASING (initial population) → Target: DECREASING by Sprint 2

---

9. Trend Analysis

9.1 Debt by Category (initial state)

Category	Items Added	Items Resolved	Net
Code Quality	2	0	+2
Testing	1	0	+1
Design	3	0	+3
Dependencies	0	0	0
Infrastructure	3	0	+3
Documentation	1	0	+1
Security	0	0	0
Performance	1	0	+1

9.2 Root Cause Analysis

Dominant category: Design + Infrastructure (3 each) **Root cause:** Prototype-speed MVP build — all design/infrastructure decisions deferred in favor of demonstrating UI **Proposed systemic fix:** For future Bilko features, CI/CD + tests must be in place before any feature PR is mergeable. Hook: PR blocked without test file alongside new service.

---

10. Resolved / Accepted Debt Archive

ID	Description	Resolution	Resolution Date	Sprint	Resolved By

—	No resolved items yet	—	—	—	—

Approval

Role	Name	Date	Signature
Author	John (AI Director)	2026-02-23	
Reviewer			
Tech Lead	John	2026-02-23	
AI Director (John)	John	2026-02-23	