Introduction to Development

At Unicis, our development process is the backbone of creating secure, accessible, and compliant solutions. This page outlines our methodology, team structure, standards, and best practices.

Last updated: 2026-06-12 — UX/UI Redesign, Dashboard & Accessibility release (Direction B design system, WCAG 2.1 AA, 7-locale i18n, responsive design).
Full testing and accessibility procedures: Testing & QA Guide

Development Philosophy and Approach

Agile, but not limited to and not strictly by the book.

Iterative, incremental, and evolutionary

Agile methodologies divide product development into sprints (1–4 weeks). Each sprint involves planning, analysis, design, coding, unit testing, and acceptance testing. This approach minimises initial planning, reduces risk, and enables the product to adapt quickly.

Efficient and face-to-face communication

The sixth principle of the Agile Manifesto stresses face-to-face conversation. Each team should have a customer representative (product owner) to answer questions and align priorities. At Unicis we use OpenProject for project tracking.

Very short feedback loop and adaptation cycle

Daily stand-ups (15 minutes) where team members review progress, plans, and impediments.

Quality focus

Continuous integration, automated testing, and test-driven development — including accessibility, performance, and internationalisation from day one.

Development Team Structure

Role	Responsibilities
Developers	Write and test code, participate in code review, meet accessibility and i18n requirements, maintain documentation.
Designers	Maintain the Direction B design system, produce responsive UI, validate WCAG 2.1 AA contrast and touch-target requirements.
DevOps	Manage CI/CD, system reliability, infrastructure, and deployment pipelines.
Test / QA	Write Jest unit tests and Playwright E2E suites, run accessibility audits, verify responsive behaviour at all breakpoints.

Development Process

Sprint Planning

Detailed planning of tasks ensuring alignment with project goals and acceptance criteria — including accessibility and responsive design criteria where applicable.

Code Review

Peer reviews to maintain high code quality. Every pull request must pass TypeScript type-checks, ESLint, and the full automated test suite before merge.

Testing

Combination of automated (Jest, Playwright) and manual testing to ensure functionality, performance, security, accessibility, and responsiveness. See Testing & QA Guide.

CI/CD Pipeline

GitLab CI (primary) / GitHub Actions (Community Edition): runs tsc –noEmit, ESLint, Jest unit tests, and Playwright E2E on every pull request. Deployment via CapRover. See DevOps for full pipeline docs.

Accessibility Standards — WCAG 2.1 AA

WCAG 2.1 Level AA is a **hard requirement** for all new and redesigned pages. Accessibility is treated as a first-class requirement, not an afterthought.

Core Requirements

Criterion	Requirement	Standard
Language attribute	Every page sets `<html lang>` from the SSR locale.	3.1.1
Colour contrast	Minimum 4.5:1 for normal text; 3:1 for large text and UI components.	1.4.3 / 1.4.11
Keyboard navigation	All interactive elements reachable and operable via keyboard.	2.1.1
Focus indicators	Visible focus ring on all focusable elements.	2.4.7
Touch targets	Minimum 44 × 44 CSS px for all interactive controls.	2.5.5
ARIA roles	Correct roles on tabs (tablist/tab/tabpanel), dialogs (dialog, aria-modal), charts (role=“img”).	4.1.2
Form labels	All inputs have label or aria-label; errors use aria-invalid + aria-describedby.	3.3.1 / 3.3.2
Modal dialogs	Focus moves to modal on open; Tab trapped inside; Escape closes; focus returns on close.	2.4.3
Icons / images	Decorative icons carry aria-hidden=“true”; informational icons carry aria-label.	1.1.1
External links	Include (opens in new tab) as sr-only text.	2.4.4

Implemented in 2026-06-12 Release

- <html lang> set from SSR locale on every page
- Chart.js canvases wrapped in role="img" with descriptive aria-label
- Mobile sidebar rewritten as accessible dialog: role="dialog", aria-modal, focus-on-open, Escape, Tab focus trap
- Dashboard tab bar: full ARIA tab pattern (tablist, tab, tabpanel)
- Notification bell aria-label is dynamic with unread count; badge carries aria-hidden
- Webhook form: fieldset+legend for events; aria-invalid, aria-describedby, role="alert" for errors
- All icon-only buttons carry aria-label; active domain health cards carry aria-pressed
- Content text colour raised to text-slate-500 across all key areas to meet 4.5:1 contrast ratio

See Testing & QA Guide → Accessibility for the full audit checklist.

Design System — Direction B

All modules use the **Direction B** design language. The `design.md` file in the repository root is the authoritative reference.

Principles

- **Calm and professional** — avoid loud colours, excessive borders, or decorative noise
- **Consistent token set** — use Tailwind tokens (slate-*, ub-blue, ub-green, ub-purple, ub-amber, ub-red); never hardcode hex values
- **Dark mode** — every component must look correct in both light and dark mode
- **Responsive by default** — verify at 375 px (mobile), 768 px (tablet), and 1280 px (desktop)

Key Component Patterns

Pattern	Guidance
Page shell	bg-white dark:bg-slate-900 rounded-2xl border border-slate-200 dark:border-slate-700
Table headers	text-[11px] font-semibold uppercase tracking-wider text-slate-500
Multi-step dialogs	Sticky footer: flex flex-col overflow-hidden on DialogContent; flex-1 min-h-0 overflow-y-auto on body div
Task picker	Use Combobox (not Select) — provides search, truncation, and Radix portal
Buttons	Shadcn Button — do not use DaisyUI btn classes in new code
Tables	Nested pattern: overflow-hidden rounded-xl outer / overflow-x-auto inner

Internationalisation (i18n)

Unicis Platform supports **7 locales**: English (en), French (fr), German (de), Spanish (es), Italian (it), Japanese (ja), Portuguese (pt). Italian, Japanese, and Portuguese were added in the 2026-06-12 release with 880+ keys per language.

Rules for Developers

- Never hardcode English strings in components — always use t('key')
- Namespace prefix syntax (t('tia:key')) requires the namespace in the hook: useTranslation(['common', 'tia'])
- Every page must list in serverSideTranslations all namespaces needed by any dialog it can open
- New keys must be added to **all 7 locale files** before the PR is merged
- Use count and status interpolation — never concatenate translated strings

Page Namespace Reference

Page	Required namespaces
rpa.tsx	common, rpa, tia, pia
tia.tsx	common, tia
pia.tsx	common, pia
csc.tsx	common, csc
risk-management.tsx	common, rm
iap/index.tsx + iap/admin.tsx	common, iap

Tools and Technologies

Layer	Technology
Framework	Next.js (Pages Router), TypeScript
Styling	Tailwind CSS v4 JIT with custom design tokens
Components	Shadcn/UI, Radix UI primitives, Lucide React icons
Data fetching	SWR (all mutate() calls must be awaited)
ORM / DB	Prisma + PostgreSQL
Auth	BoxyHQ (SSO/SAML)
i18n	next-i18next (7 locales)
Unit tests	Jest
E2E tests	Playwright
CI/CD	GitLab CI · GitHub Actions · CapRover

See DevOps for full pipeline documentation.

Documentation

Importance of Documentation
Ensures clarity, facilitates onboarding, and maintains consistency.

- **API docs** — maintained in openapi/openapi.yaml, regenerated to public/openapi.json via scripts/generate-openapi.ts
- **Design system** — design.md in the repository root is the authoritative Direction B reference
- **Handbook** — this wiki; updated with every major release

Best Practices and Standards

Coding Standards

No hardcoded English strings. No inline hex colours — use design tokens. No backwards-compatibility shims for removed code. No fire-and-forget mutate() calls — always await.

Accessibility

WCAG 2.1 AA is a hard requirement. Run the accessibility checklist before every pull request.

Responsive Design

Verify at 375 px, 768 px, and 1280 px before merge. Tables must scroll horizontally; dialogs must never overflow; toolbars must stack on narrow screens.

Security Practices

Encryption, access controls, and regular security audits. We follow MVSP to prioritise essential security measures from the earliest stages of development.

Additional Resources

- 12 Factor App methodology
- Minimum Viable Secure Product (MVSP)
- WCAG 2.1 Quick Reference
- Unicis Testing & QA Guide

Release Process

Pre-release Checklist

1. [ ] New features have Jest unit tests; API endpoints have handler-level tests
2. [ ] E2E Playwright test added for critical user flows
3. [ ] WCAG 2.1 AA checklist passed (see Testing Guide)
4. [ ] Verified at 375 px, 768 px, and 1280 px viewports
5. [ ] All 7 locale files updated for new translation keys
6. [ ] npx tsc –noEmit –skipLibCheck passes with no errors
7. [ ] CHANGELOG updated

Release History

Release	Date	Highlights
UX/UI Redesign, Dashboard & Accessibility	2026-06-12	Direction B design system, WCAG 2.1 AA, IT/JA/PT locales added (880+ keys each), 62 Jest tests, 3 Playwright E2E suites, responsive design across all modules, KPI dashboard row, billing redesign
Tasks	2026-05-31	Task import templates from CSC controls, Kanban board, CSC control status dialog

Training and Development

New developers must read design.md (Direction B design system) and Testing & QA Guide before their first PR. Continued education through training programmes covering accessibility standards, new platform features, and industry best practices.

Feedback and Improvement

We collect feedback through surveys, usability tests, and direct user interaction. Retrospectives, metrics analysis, and proactive problem-solving drive continuous improvement in efficiency, quality, and customer value.

agile, development, best practice, framework, training, CI/CD, sprint, DevOps, code review, testing, accessibility, WCAG, i18n, responsive, direction-b, design-system

From:
https://handbook.unicis.tech/ - **Unicis Handbook**

Permanent link:
https://handbook.unicis.tech/pub/development?rev=1781245503

Last update: **12.06.2026 06:25**