

# Technical Health Check

Stay ahead of the competition.

## Performance Acceleration

Get a technical health check to **increase your team's performance**. We watch for destructive patterns and architecture pitfalls and check your team's technical skills against your product roadmap requirements.

## Improve Conversion Rates

Improve your customer's experience by providing solid and blazing-fast applications that scale. Fast and snappy applications result in **better user satisfaction** and **higher conversion rates**.

## Reduce Costs

Get the best out of your budget. Run a technical health check to find architectural problems and knowledge gaps in advance and **reduce** long-term **bug fixing, maintenance,** and **support costs**.

## Increase Product Quality

Get a technical health check to improve your product's quality, stability, and maintainability, to **ship** further versions **quickly and stable**.



# Technical Health Check

We run the audit your **Angular** application deserves.

## Why Health Checks?

- ✓ Get a summary of the current tech state of the project
- ✓ Get expert advice before significant transitions
- ✓ Transform your code and team for future proof
- ✓ Scan your code base for unknown problems
- ✓ Get an unbiased opinion about your project
- ✓ Align your tech expertise with your product roadmap
- ✓ Improve team collaboration and knowledge sharing
- ✓ Empower your team to get the best results
- ✓ Close the check with a motivated team

## Technology Focus

We have a strong expertise in the Angular ecosystem including nx mono repositories, tooling and NestJS for full-stack applications.



## qupaya, your unicorns.

- ✓ Expertise since the early days of the web
- ✓ Straight focus on one technology ecosystem
- ✓ Strong GDE and core developer network
- ✓ Open-source and Angular community contributor
- ✓ We work with customers of any size and industries
- ✓ A global team of experts and architects

## Expertise From Ground Up

- ✓ HTML Parsing
- ✓ CSS Parsing
- ✓ JavaScript Execution
- ✓ Angular Compilation
- ✓ Runtime Execution
- ✓ Change Detection
- ✓ Rendering
- ✓ Painting
- ✓ Event Handling



# Phases

We evaluate. Step by step.

## I Onboarding

During the first days, we meet the management and responsible developers to define the primary needs and address relevant problems. A dedicated counterpart and software architect will work closely with you during the check.

We better understand your product, the big picture, the roadmap, and the way you work and collaborate. We evaluate the target group of your customers, devices, and performance metrics and determine the software usage.

## II Code Review

The code review is central to your application's health check and takes the most time to understand your application's architecture and coding patterns in detail. We start with an automated analysis of your project and dependencies to get a quick overview of the code base. Afterward, the responsible architect will lead the review and consult experts on specific topics.

We analyze your application against many topics, such as architecture patterns and coding paradigms, data management, testing principles, component, UI design, etc.

## III Team Review

If desired, we go into an exchange with your team members for technical interviews to determine their expertise, motivation, and strengths. We create an overview of the technical knowledge and learning opportunities. You get a report including knowledge gaps and skills in your unit and advice on collaboration and learning techniques to strengthen and share the development skills of your team members.

## IV Offboarding

During the last days, we finalize and present the results, advise recommendations for actions and suggest possible changes in the technical roadmap. In addition, we offer pair and mob programming and hands-on sessions to solve the most relevant and complex issues.

You get additional content and educational information for self-learning, meetups, conferences, and workshops.

**However, you get a motivated team with straight action points to get the best out of your Angular application!**



# Technical Topics

We cover your needs.

## Technology Stack and Libraries

We analyze the project dependencies and run plausibility checks not covered by package managers, such as NPM. You'll get a report about outdated, unmaintained libraries or dependencies lacking long-term or community support. We help you reduce dependencies and create a migration path if needed. Due to our community network, we advise bleeding-edge libraries and technologies just around the corner.

The more rarely-used dependencies, the less the hiring pool with adequate candidates.

## Search Engine Optimization

Dynamic applications offer a wide range of flexibility and performance improvements but lack SEO support. However, with the correct architecture, your single-page application can ship perfectly crawlable for search engines like Google, Bing, or DuckDuckGo.

We check your current application architecture and will if needed, advise a migration path to set up server-side rendering on the on-the-fly or in advance. We care about cloaking issues and will improve user experience through pre-rendered responses and application hydration.

## Architecture & Paradigms

The software architecture review is an essential part. It involves evaluating the high-level design of your application and ensuring that it aligns with the business requirements, meets performance expectations, and is maintainable, scalable, and secure. In addition, a thorough architecture review will identify design weaknesses or potential issues that could compromise the software's functionality or quality.

One of the critical areas of the architecture review cover is the choice of architectural patterns. Different architectural practices, such as Layered Architecture, Module Federation Architecture, Service-Oriented Architecture (SOA), Event-Driven Architecture (EDA), and Domain-Driven Design (DDD), have their advantages and disadvantages, and the choice of the pattern should be based on the specific needs of the project.

In conclusion, the architecture review is a crucial step in ensuring the success of a software development project. By evaluating architectural patterns, data model design, security, scalability, and performance, developers can create a robust and effective software system that meets the business's and end-user's needs and solidifies your project for long-term maintenance.



# Technical Topics

We cover your needs.

## Component Design (Semantic UI)

Component design is a crucial aspect of building robust and maintainable Angular applications. When designing components, it's essential to consider how to structure UI components, create semantic custom components, and design reusable layouts and components.

Designing custom components with a clear purpose and use makes understanding and modifying the application's behavior easier. Semantic elements also promote consistency and make it easier for developers to collaborate, reducing the likelihood of errors and bugs.

Creating reusable structures makes maintaining consistency throughout the application and avoiding unnecessary code duplication easier. Reusable components also make adding new features to the application more accessible, reducing the time and cost of development.

We help you to follow these component design principles to build robust and maintainable Angular applications that can be easily modified and scaled. Also, these practices promote collaboration among developers, leading to more efficient development and a better end product.

## Data Management

Data flow, data management, and state management are critical to building robust and scalable applications. In today's world of data-driven applications, it's essential to clearly understand how data flows through the application, how it's managed and stored, and how to maintain consistency across different components.

We analyze your software against common data management paradigms, such as client-side data flow, storage, and synchronization. We also advise a data handling pattern that scales with your code and team. A well-defined and documented data flow enables developers to onboard quickly in large software projects.

## Code Quality

We run a complete static code analysis and a manual sample review of your code base to check against code complexity, maintainability, and readability. We advise shared code styles that fit the needs of your development team, lower the learning curve for new developers and reduce git conflicts.



# Technical Topics

We cover your needs.

## Testing Principles

If you're looking to build a top-notch application, software testing is an essential step to ensure the quality and reliability of your code. Testing not only helps to catch bugs and errors but also helps to prevent issues from arising in the first place. With thorough testing, you can be confident that your application will function as intended, delivering an optimal user experience.

At the heart of software testing is detecting and addressing issues before they impact your users. By performing both unit and integration tests, you can ensure that each component of your Angular application functions correctly and works seamlessly with the rest of the system. This can save you time and resources by identifying problems early in the development process and allowing you to make necessary changes before they become more complex and costly to fix.

In addition, software testing can help you to maintain the security and privacy of your users' data. By thoroughly testing your application, you can ensure it is protected from vulnerabilities and potential security threats.

Overall, software testing is a critical component of any successful development project. By prioritizing testing in your development process, you can be confident in the quality and reliability of your code and let your team sleep well.

## CI / CD

Continuous Integration (CI) and Continuous Delivery (CD) are practices for modern software development teams. CI ensures that code changes are frequently integrated into a central repository and tested automatically, preventing integration issues that can delay project delivery. CD automates the entire software release process, allowing teams to deploy code changes to production in minutes, not days or weeks.

By implementing CI and CD, software development teams can deliver high-quality software faster and more efficiently. These practices also enable teams to detect and fix issues early in the development process, reducing the risk of critical bugs and ensuring that the software meets the needs of its users.

We help you to automate your workflows with Angular, Nest, or nx workspaces to save time and increase software quality.



# Technical Topics

We cover your needs.

## Security

We check your application against common vulnerabilities by static code analysis, a manual code review, and during a runtime test. The study covers the front (Angular, Browser Security Policies), transportation layer (HTTP, WS), and backend parts (NestJS/Express) if needed.

Your team will get advice on minimizing dependencies and reducing access and privileges to a minimum.

We run checks, for example, including cross-site scripting (XSS), code injections, authentication, and authorization issues. We also advise resource identity checks and the possibility of resource restrictions by the CORS pattern.

## GDPR and Data Processing

Personalized data is gold and must be stored and processed carefully. Complex applications using third-party resources need regular compliance check-ups to avoid legal pitfalls.

We help you identify possible issues and advise on minimizing personalized user data and external capabilities.

## Performance

Performant and snappy applications improve conversions, increase user satisfaction, and increase SEO ranking. Also, reduced bandwidth and innovative caching strategies lessen the carbon footprint and costs of your hosting infrastructure of apps with large user bases and heavy internet traffic. We analyze your application to enhance the load, view, and interact time (TTL (Time To Load) / TTV (Time To View) / TTI (Time To Interact)).

Also, if possible, we suggest optimistic rendering to improve our application's response to user interactions. During runtime checks, we analyze static resources delivered by CDN or inline and validate the server response for caching strategies. Our architects watch out for long-running and heavy scripts that can be optimized by rendering improvements, web-, and service workers.

The check will consider your audience and target devices, such as desktop and mobile versions, and users with flaky internet connections or weak hardware.



# Technical Topics

We cover your needs.

## Internationalization & Localization

Localization (i10n) and internationalization (i18n) of applications are more than simple text replacements for different locales. We recommend a solid technical foundation, even in the early technical planning phase, to make your internationalization scalable in the long term. Late implementations result in complex and long-running refactorings that affect almost every part of your software.

We help you to achieve localization and internationalization at any stage of your project and advice for international standards, such as ICU (International Components for Unicode), a web design pattern for RTL/LTR content, currency and date formats, sellouts, timezone handling and many more.

## Accessibility (a11y)

We validate your front end against legal requirements, such as the European Directive for Web Accessibility, and advise on how to improve your user experience for persons, with disabilities, for example, such as one-handed or (color-) blind persons.

Besides user happiness, accessible websites and applications can improve your SEO ranking.

## Reactive Programming (RxJS)

RxJS is a library for reactive programming using Observables, which can be used in both Angular and NestJS contexts. In Angular, RxJS is used extensively for managing asynchronous data streams, such as HTTP requests or user input events. Using RxJS, developers can easily compose and transform these data streams to create responsive and reactive user interfaces.

In NestJS, RxJS handles reactive programming within the Node.js environment. NestJS provides a built-in module for RxJS, which can be used to build reactive APIs and services. As a result, developers can leverage the power of RxJS to handle asynchronous operations, data streams, and events in a more efficient and scalable way.

During the health check, we check your code against memory leaks and the correct and intended use of data streams; watch out for deprecated operators and creator functions and advise for custom mono and multi-operators to simplify your streams.

Even with the upcoming Signals feature, we suggest migration paths keep your changes minimal but your application reactive.



# Technical Topics

We cover your needs.

## Progressive Web Apps (PWA)

Progressive Web Apps (PWAs) are web applications that use modern web capabilities to deliver an app-like experience to users. PWAs can run on any device with a modern browser and can be installed on the user's device without going through an app store. This allows PWAs to offer a seamless and fast user experience.

We check the offline-first capability of the PWA. This means ensuring the PWA can still function even when the user's device is offline. During the review, we test the PWA's offline capabilities and provide a seamless user experience even without an internet connection.

The check also focuses on which resources should or should not be cached for offline use. We evaluate which resources can be cached and which should not, as caching unnecessary resources can affect the PWA's performance and user experience.

Update strategies are another critical topic evaluated during the code review. Our consultancy team checks that the PWA can be updated seamlessly without disrupting the user experience. This includes assessing how the PWA handles updates and whether it provides notifications to users.

Also, we consider the PWA's use of native browsers and device APIs and notifications.

## Hybrid Web Applications

Ionic and Electron are two popular frameworks for building cross-platform applications, including web and hybrid apps for mobile and desktop. With this technology, you can ship native-like web applications to stores and gain even more native access to the user's device.

We check your application against the bad and good patterns and advise best practices. We also recommend shipping and updating strategies, as well as a plausibility check if native deployments are necessarily or maybe can be replaced by progressive enhancements using the PWA pattern.

## Tooling & Configuration

The Angular and NestJS ecosystem can be extended with various tools and configurations to make the development work a breeze and develop stability. Depending on the project setup, we advise a well-established and proofed toolset that can be gracefully integrated into your current workflow.

