

Bronscore Quality Checklist

For high quality application development

Code quality

- Imperative and declarative code is separated
- Functional style code is used where possible
- Asynchronous functions are used where possible
- Environment variables are used and documented
- Consistent_casing is applied across the tech stack
- Typing is applied comprehensively
- Automated linting and formatting is applied
- Line lengths and function size are minimized
- Design patterns are only used if indispensable
- Algorithms, types and large functions contain comments

Application interactivity

- All interactions (except for data fetching) take at most 100ms to perform or show a loading state
- All endpoints load within 100ms except for a well documented list of reports
- The initial load of the application takes at most 1000ms
- All re-renders result in a visible change and do not affect more than the change
- Appropriate accessibility is applied
- The application is causally closed: all data entered in the application can be altered and deleted within the application.
- Fetching data or performing slow actions show a loading state
- Errors are caught and an error page is displayed
- Tables exceeding 100 rows are paginated
- Reloading the application retains the last viewed page
- Edits can be cancelled
- All requests are cached
- All tables can be sorted

Flexibility

- The ontology of application is clear for all stakeholders
- Minor ontological changes take less than 15 minutes to implement
- A migration system is set up and applies migrations automatically

Agile development

- There are regular sprint meetings together with the client
- No ad hoc changes are pushed to production
- A testing procedure is in place
- A system for acceptance testing is in place
- Retrospectives are regularly planned
- User stories of critical paths are defined
- A central project board is used to plan and track development
- Code reviews are planned regularly

Commercial and legal

- The project proposal, including scope, timeframe and costs are accepted
- A service level agreement is made
- A data processing agreement is made
- Stakeholders are updated about project progress on a regular basis
- Compliance checks are made regularly
- Intellectual property rights are clearly defined

Development experience

- Continuous integration is implemented
- Continuous deployment is implemented
- Git features are used
- The test and staging environments are online
- Setting up the entire development environment including test data takes at most one hour
- High quality test data is available

Implementation in organization

- Functional management of the application is transferred to the client
- Functional documentation is present
- The client can make reports
- Specialized reports need to be created by the development team at most once a month
- End users are trained
- Proactive monitoring of usage is present

Privacy and security

- A data protection impact assessment is made
- User activity is logged on an endpoint basis
- Fail2ban is installed on all servers
- A rollback test is performed each year
- Backups are made automatically
- Compartmentalization is applied for complex applications
- The permission model is clear and communicated and can automatically be retrieved from the system
- Updates are performed regularly

Maintainability

- Outside connections are mapped
- Software architecture is mapped
- Dependencies are listed
- Dependencies are regularly checked for updates
- Technical documentation is present
- An OpenID specification is automatically generated for API's