

## Objective

As a team, we conducted a code review using SonarQube and code coverage metrics, also for proper practice we used a default ESLint configuration and a common formatting and naming etiquette. The findings were meant to increase maintainability and readability of code.

## Summary

Tool	Report
Code Coverage	71.1%
SonarQube - Reliability	B - 110 open issues
SonarQube - Maintainability	A - 153 open issues
SonarQube - Duplications	0.0%
SonarQube - Security	A - 0 open issues

## Common & Priority Issues

### Reliability:

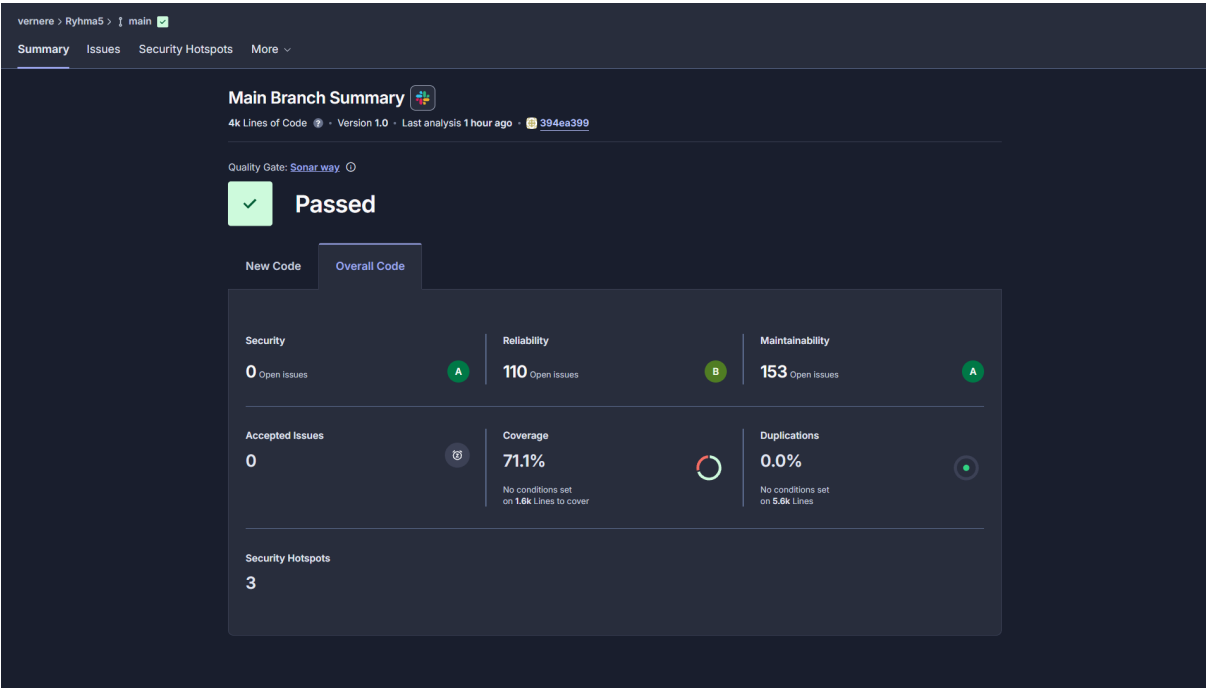
1. Missing prop validation
  - Reliability and maintainability low to medium priority
  - When passing props to components it is good practice to validate the existence of of the prop
2. Visible, non-interactive elements with click handlers must have at least one keyboard listener.
  - Reliability low priority
  - None interactive elements are not meant to have click handlers, causes readability issues

### Maintainability:

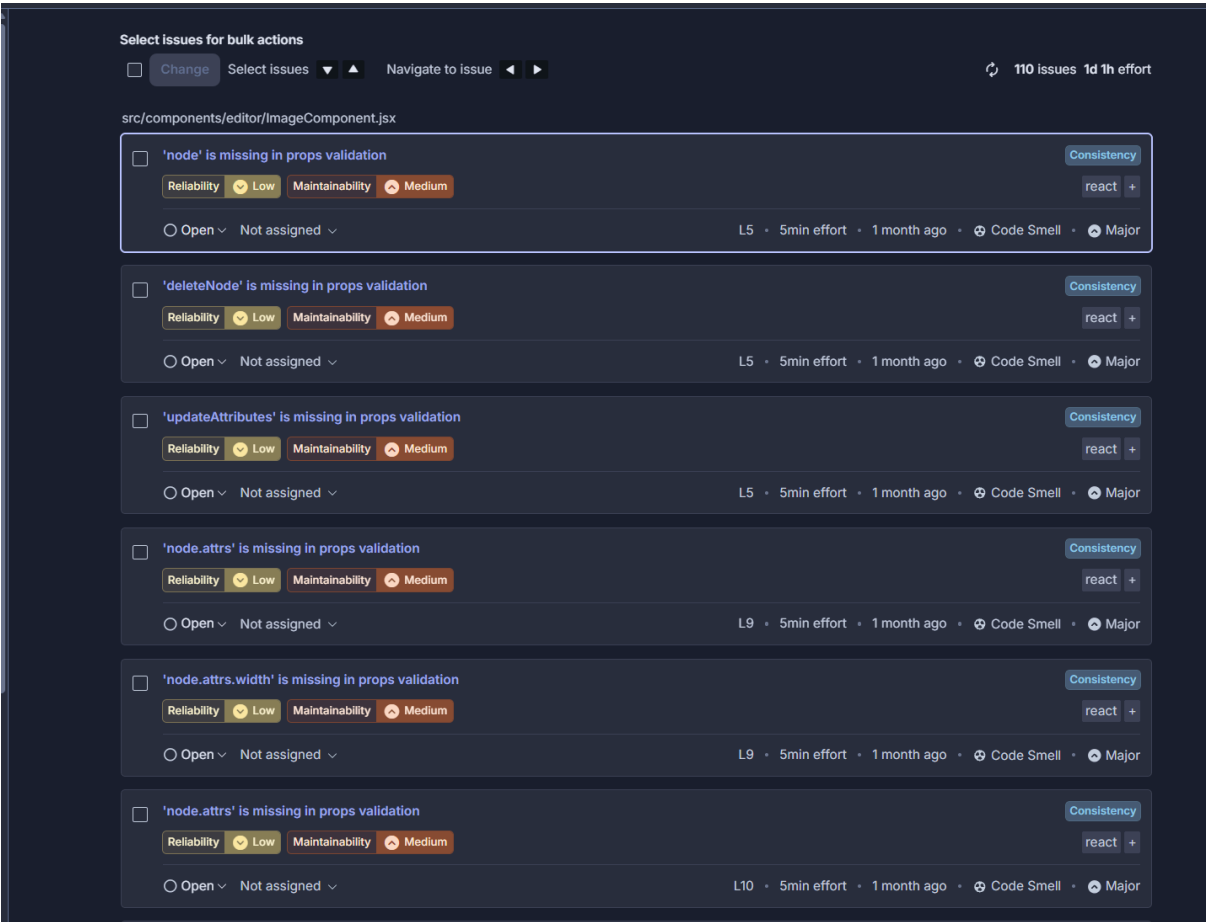
1. Components imported multiple times
  - Maintainability low priority
  - Affects readability
2. Multiple checks instead of optional chaining (?.)
  - Maintainability medium priority
  - Makes code more readable and concise
3. Unused imports
  - Maintainability low priority
  - Takes up less space

Examples:

## 1. SonarQube report



## 2. Prop validation



### 3. Missing non interactive elements

Open Not assigned L64 5min effort 1 month ago Code Smell Major

'node.attrs.title' is missing in props validation

Consistency

Reliability Low Maintainability Medium

react +

Open Not assigned L64 5min effort 1 month ago Code Smell Major

Avoid non-native interactive elements. If using native HTML is not possible, add an appropriate role and support for tabbing, mouse, keyboard, and touch inputs to an interactive content element.

Consistency

Reliability Medium Maintainability Low

accessibility react +

Open Not assigned L85 5min effort 1 month ago Code Smell Major

src/components/editor/ImageUploadButton.jsx

'editor' is missing in props validation

Consistency

Reliability Low Maintainability Medium

react +

Open Not assigned L6 5min effort 1 month ago Code Smell Major

'editor.chain' is missing in props validation

Consistency

### 4. Example error

80 81 82 83 84 85

```
<Trash2 size={18} />
</button>
</div>

/* Resize handle */
<div
```

Avoid non-native interactive elements. If using native HTML is not possible, add an appropriate role and support for tabbing, mouse, keyboard, and touch inputs to an interactive content element.

```
      className={
        `absolute bottom-0 right-0 w-4 h-4 bg-blue-500 border-2 border-white rounded-tl-lg cursor-nw-resize opacity-0 group-hover:opacity-100
        transition-opacity ${
          isResizing ? 'opacity-100' : ''
        }`
      }
      onMouseDown={handleMouseDown}
      title="Resize image"
    >
      <Move size={12} className="absolute top-0 left-0 text-white" />
    </div>
  </div>
</NodeViewWrapper>
);
}
```

### 5. Sample solution

Consistency Not conventional

Avoid non-native interactive elements. If using native HTML is not possible, add an appropriate role and support for tabbing, mouse, keyboard, and touch inputs to an interactive content element.

Non-interactive DOM elements should not have an interactive handler [javascript:S6848](#)

Software qualities impacted: Reliability Medium Maintainability Low

Open Not assigned Code Smell Major

Where is the issue? Why is this an issue? How can I fix it? Activity More info

Open in IDE

The simplest and most recommended way is to replace the non-interactive elements with interactive ones. If for some reason you can't replace the non-interactive element, you can add an interactive `role` attribute to it and manually manage its keyboard event handlers and focus. Common interactive roles include `button`, `link`, `checkbox`, `menuitem`, `menuttencheckbox`, `menuttenradio`, `option`, `radio`, `searchbox`, `switch`, and `textbox`.

Noncompliant code example

```
<div onClick={() => {}} />; // Noncompliant
```

Compliant solution

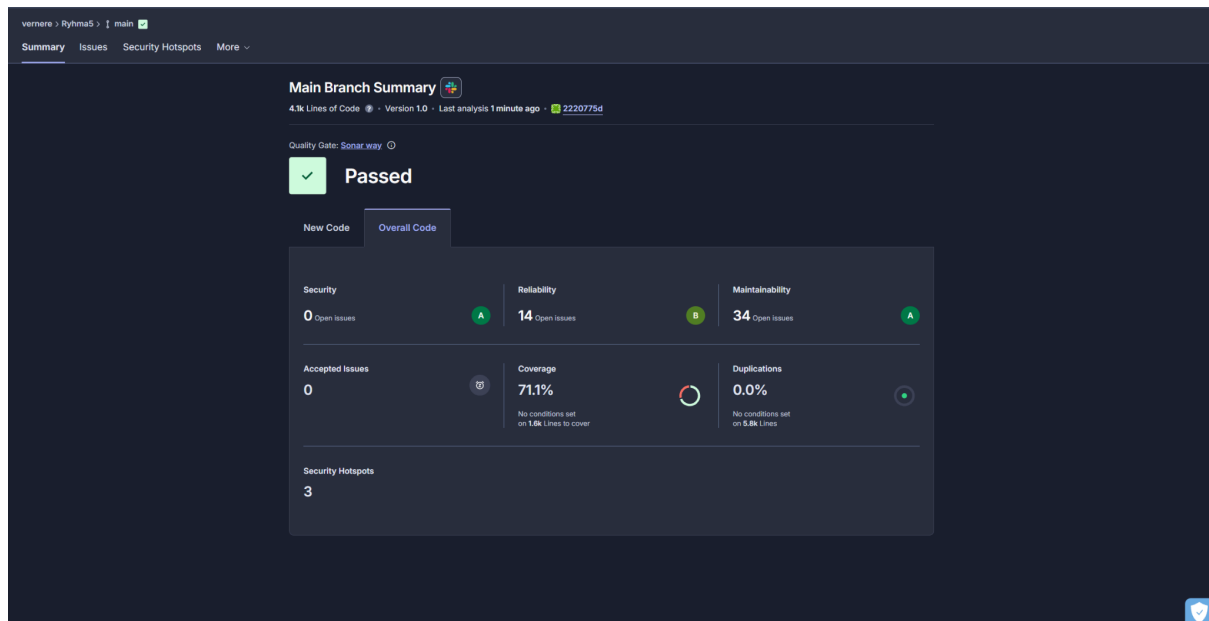
```
<div onClick={() => {}} role="button" />;
```

### Short-Term Actions(completed):

- Added prop validation to all components that get passed props
- Upped code readability through removing duplicate imports
- Changed non-interactive elements with click handlers to have at least one keyboard listener

### Long-Term Improvements:

- Refactor collaboration invitation logic for better reliability



### Conclusion:

From the beginning the code was designed and implemented with clear intent. As a result the code quality was reasonably good starting the cleanup. With the cleanup and refactoring, issues were reduced from 263 to 48 ( 81.75% reduction), which can be considered a successful result. These corrections improved the software's reliability and maintainability.