

A Brief Note on Component Design and Technical Debt

Shaan Fulton

June 29, 2025

1 A Brief Note on Technical Debt

Thoughtful, well-structured React component design enables development to scale efficiently and sustainably. If every new feature requires building a component from scratch, your development effort grows linearly with the number of additions. In contrast, by establishing reusable building blocks, you can compose new components quicker the more building blocks you implement, reducing the time needed for each new feature and allowing development to grow almost logarithmically. This approach is essential for scaling large projects with ease.

The following video provides excellent insights into strategies for avoiding technical debt and creating this form of logarithmic development time in React. Here are some of the core takeaways:

- Use context and avoid prop drilling in most situations.
- Favor a “children-first” design—leverage **children** instead of creating endless props, making components more flexible and customizable.
- Focus on developing robust building blocks rather than tightly-coupled, purpose-built components.
- Maintain a high-level perspective: always anticipate future needs, ensuring your current decisions will remain adaptable as your project evolves.

Watch the video [here](#)