

# Forces et Faiblesses de React selon les Cycles de Développement

Cycle de développement	Forces de React	Faiblesses de React
MVP (Lean Startup)	Component-based, rapide à prototyper, écosystème riche (Next, Vite, etc.)	Nécessite parfois trop de choix techniques (router, state, etc.) pour MVP
Big Design Up Front (BDUF)	Réutilisation possible de composants dans une architecture bien pensée	Pas orienté architecture formelle ou modélisation UML
Waterfall / Cycle en V	Bonne intégration dans les étapes de tests UI, Storybook, Jest	Liberté de structure mal adaptée à une cascade rigide
Architecture-First / Model-Driven	Supporte la séparation des responsabilités via hooks/custom patterns	Demande un travail de structuration non imposé par défaut
Scientific / Hypothesis-Driven	Mock UI pour tests de modèles ML ou d'algorithmes avec Storybook	Pas pensé pour la vérification formelle ou scientifique
Design Thinking / UX-First	Facile à prototyper avec des designers, forte adoption UX	Couche technique parfois trop présente pour des tests purement UX
Contractualisation / Spécifications	Peut être utilisé comme moteur de rendu dans une architecture dirigée contrat	Manque de formalisme pour répondre à des clauses contractuelles précises
Release-Driven / Milestone Planning	Excellente gestion des versions via mono-repo et composants indépendants	Pas de structure native de release planning sans tooling externe
No-Code / Low-Code	Support de bibliothèques UI sans code, intégrable via iframe/embedding	Pas destiné à générer directement des applications sans développeurs
Pretotyping / Fake Door	Landing pages interactives faciles à générer avec React SPA	L'illusion d'un produit fonctionnel peut être trompeuse si mal géré