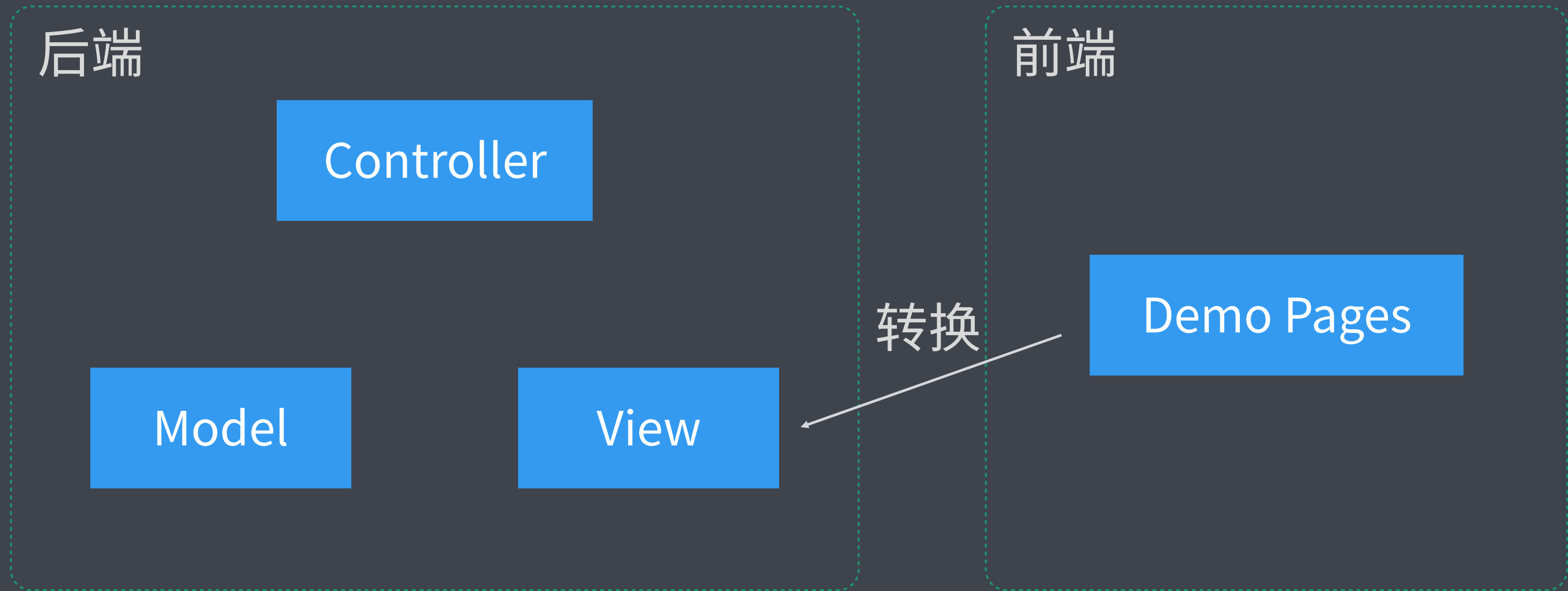


前端工程化

全副武装：通过工程化提升「战斗力」

「刀耕火种」的年代



面临的问题

技术是为了解决问题而存在的

想要使用 ES6+ 新特性，但是兼容有问题

想要使用 Less / Sass / PostCSS 增强 CSS 的编程性
但是运行环境不能直接支持

想要使用模块化的方式提高项目的可维护性
但运行环境不能直接支持

部署上线前需要手动压缩代码及资源文件
部署过程需要手动上传代码到服务器

多人协作开发，无法硬性统一大家的代码风格
从仓库中 pull 回来的代码质量无法保证

部分功能开发时需要等待后端服务接口提前完成

主要解决的问题

传统语言或语法的弊端

无法使用模块化 / 组件化

重复的机械式工作

代码风格统一、质量保证

依赖后端服务接口支持

整体依赖后端项目

工程化表现

一切以提高效率、降低成本、质量保证为目的的手段都属于「工程化」

一切重复的工作都应该被自动化



一切重复的工作都应该被自动化

- 创建项目结构
- 创建特定类型文件

创建项目

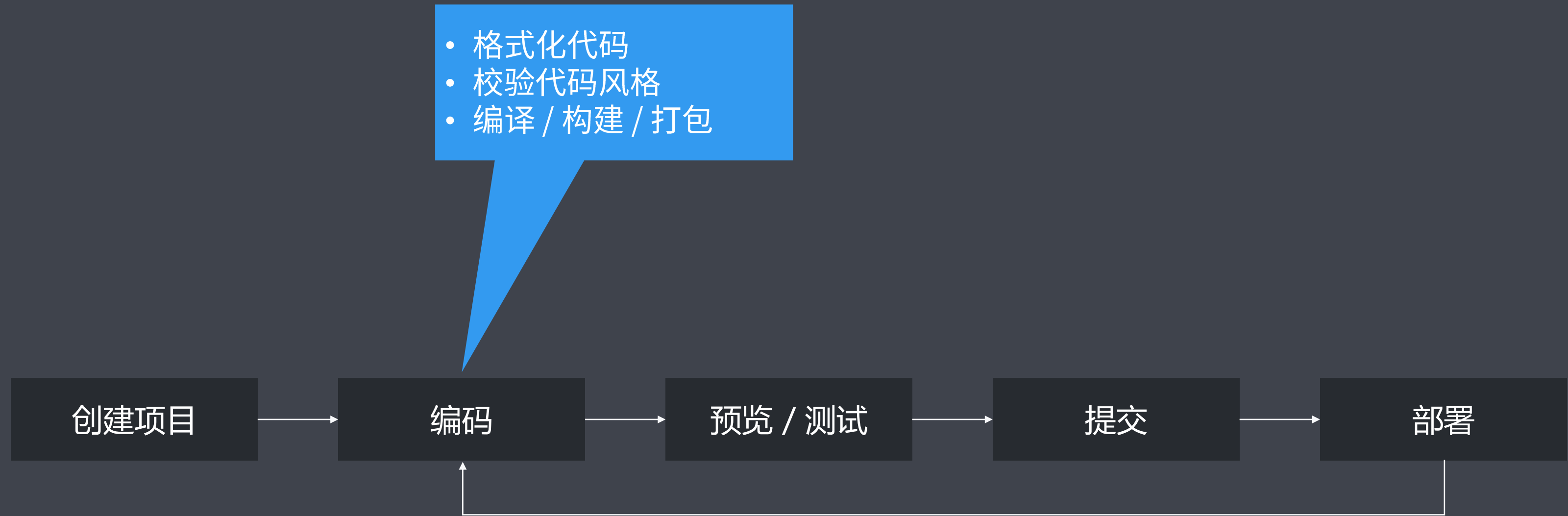
编码

预览 / 测试

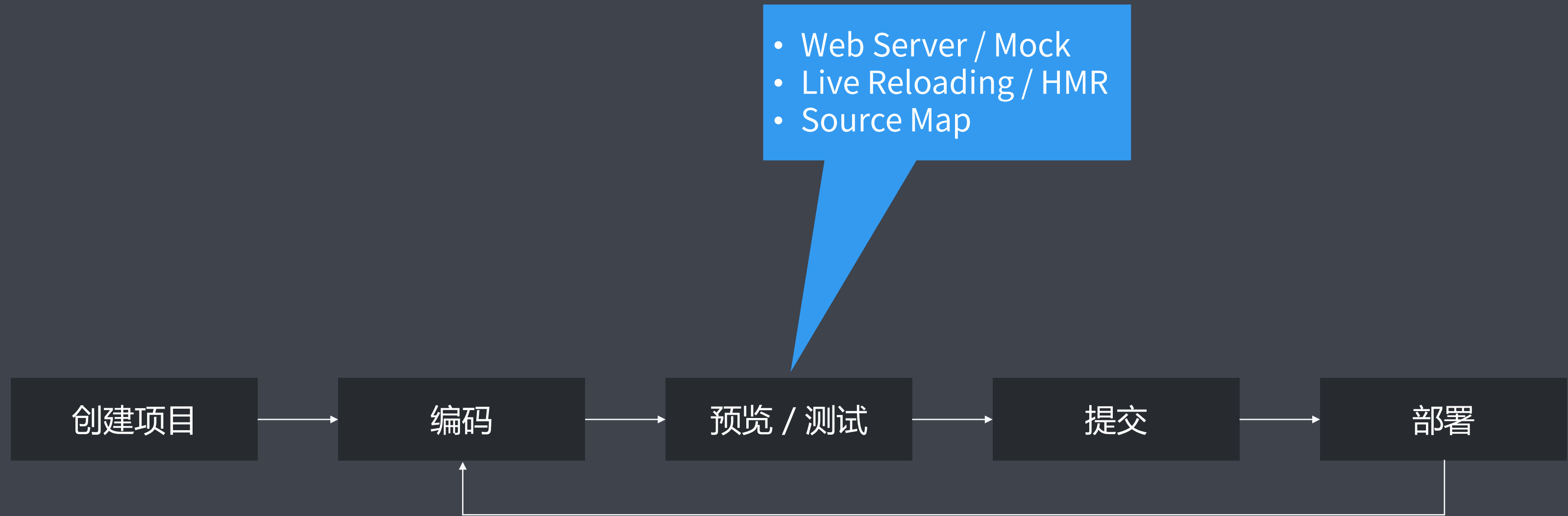
提交

部署

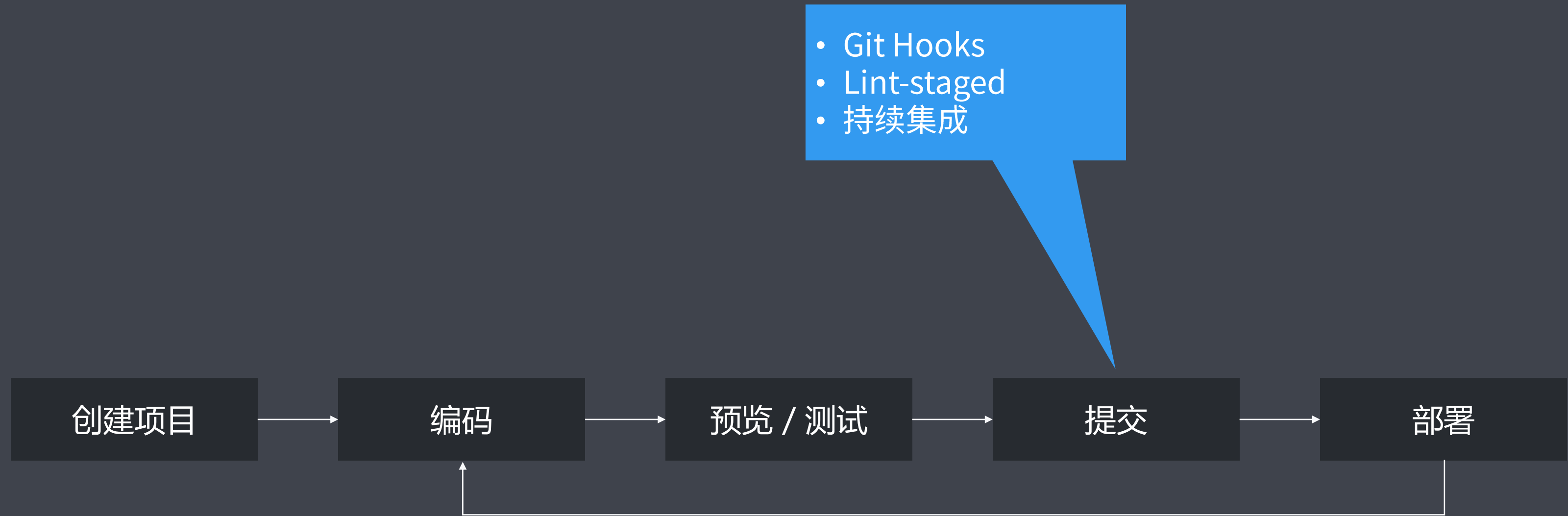
一切重复的工作都应该被自动化



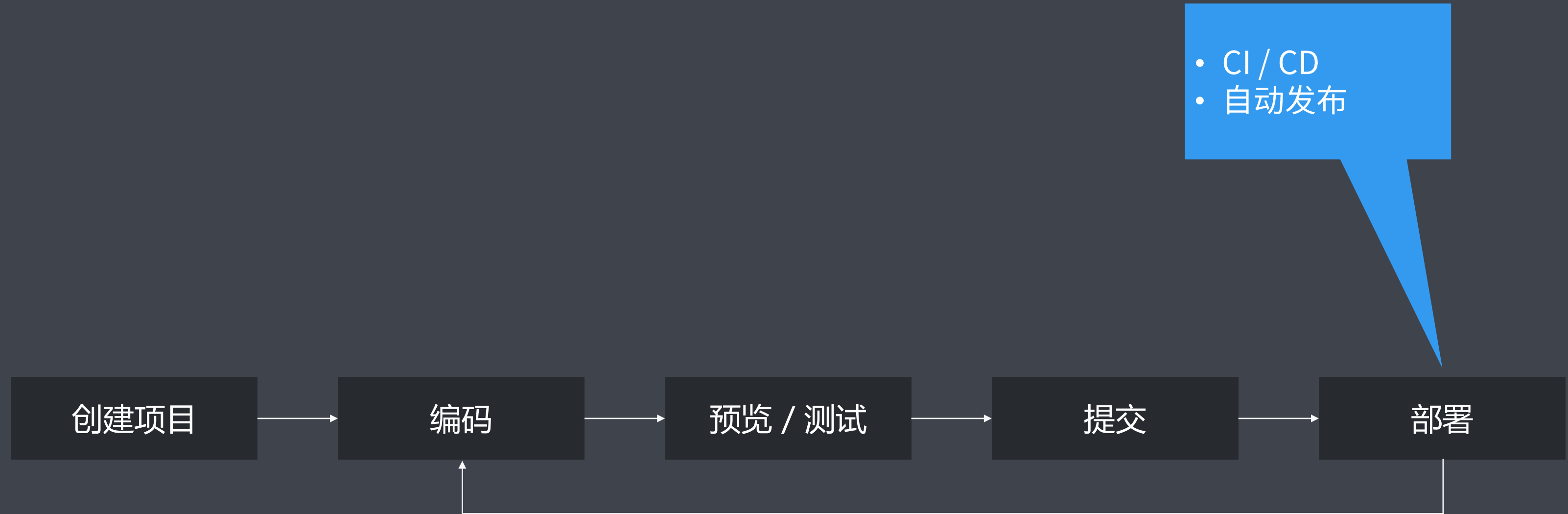
一切重复的工作都应该被自动化



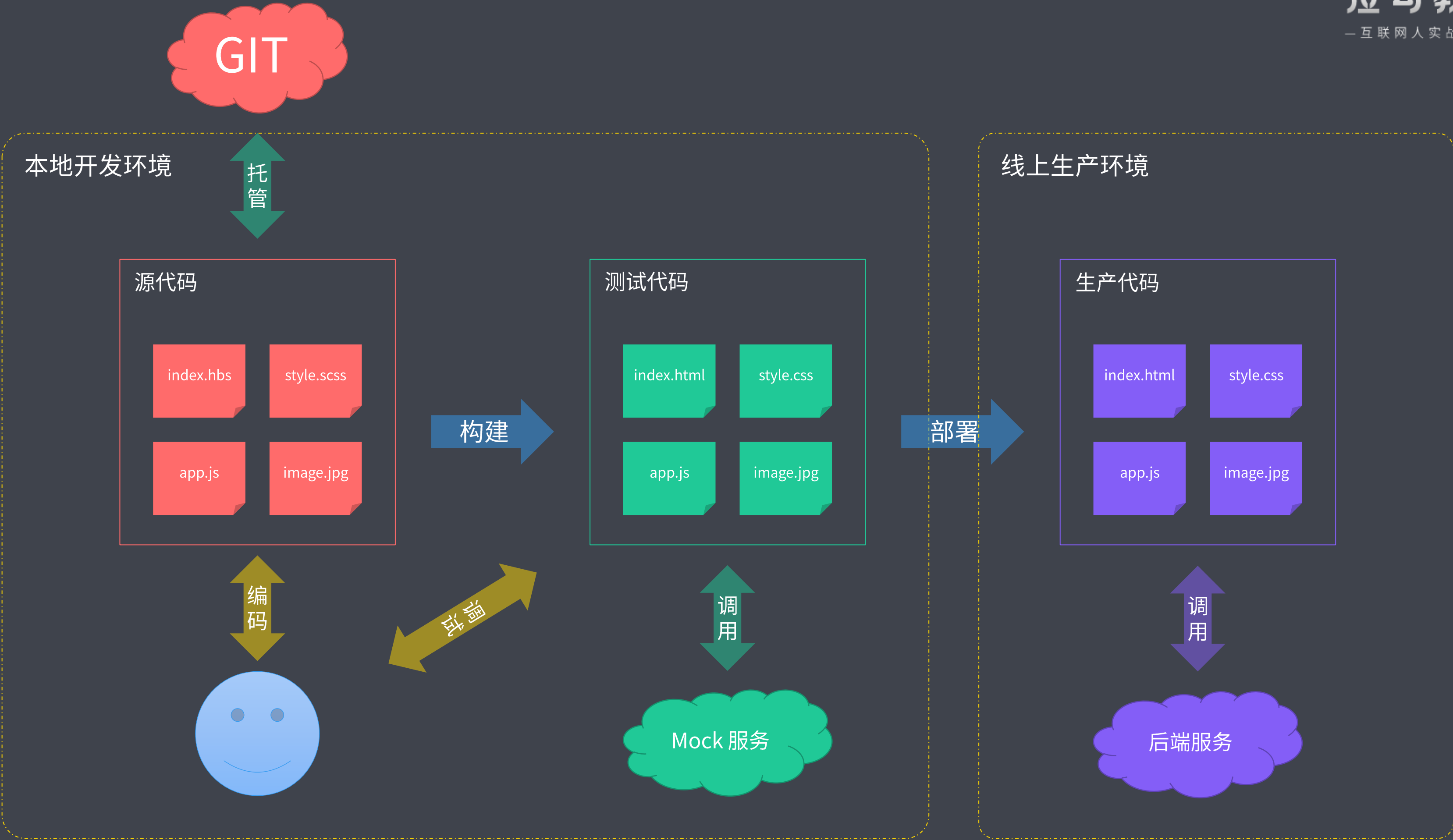
一切重复的工作都应该被自动化



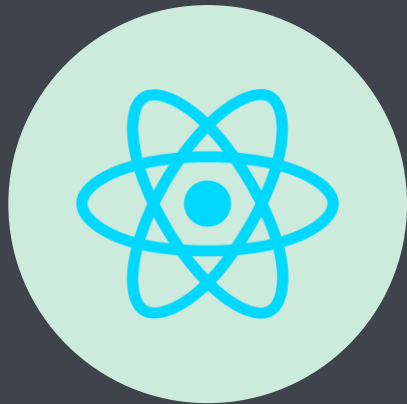
一切重复的工作都应该被自动化



工程化 \neq 某个工具



一些成熟的工程化集成



create-react-app



vue-cli



angular-cli



gatsby-cli

Powered by Node.js

厥功至伟的 Node.js

内容概要

SUMMARY

- 脚手架工具开发
- 自动化构建系统
- 模块化打包
- 项目代码规范化
- 自动化部署