

GitHubを中心とした最新開発スタイルの導入

～ 設計からデプロイまでを一元管理 ～

ITエンジニア向けライトニングトーク

Presented by: @tomotoaki

目次

1. GitHubの進化と現状
2. AsciiDocによる設計書管理
3. コードレビューの自動化
4. GitHub Actionsで実現するCI/CD
5. モダン開発フローの全体像
6. 導入メリットとTips
7. まとめ・Q&A



GitHubの進化と現状

- 従来: ソースコードのバージョン管理
- 現在:
 - ✓ 設計書管理
 - ✓ コードレビュー支援
 - ✓ CI/CDパイプライン
 - ✓ チームコラボレーション

GitHubは「単なるGitホスティング」ではない。



AsciiDocによる設計書管理

- AsciiDocとは？
 - プレーンテキストで構造化文書を記述できるマークアップ言語
- Git管理に最適：変更履歴・差分が一目瞭然
- GitHub上でプレビュー表示が可能

📌 使用例:

```
= 設計書タイトル
:author: エンジニア名
:toc:
```

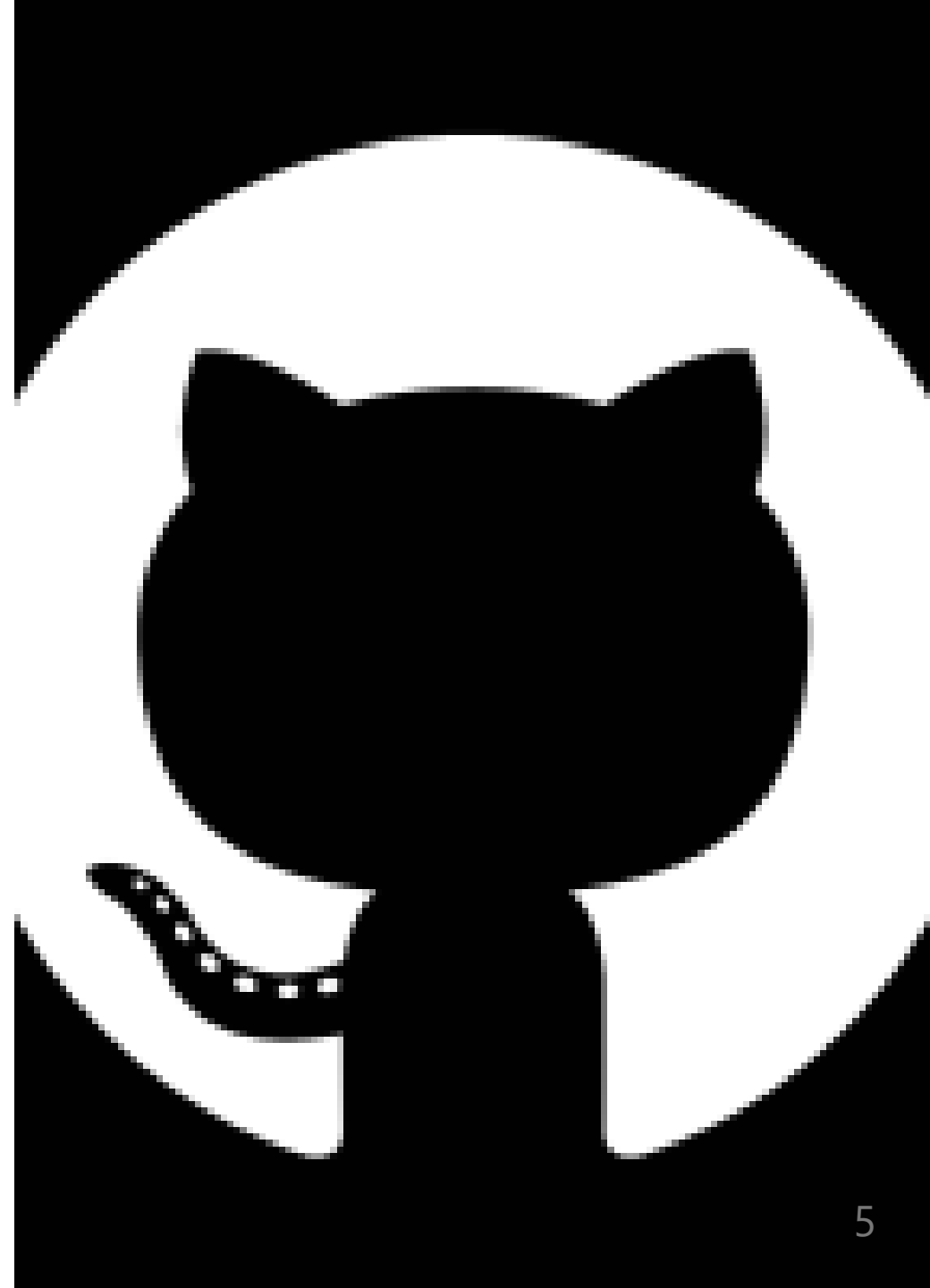
```
== システム概要
本システムは...
```

コードレビューの自動化

Pull Request + GitHub Actions

- PR作成時に自動Lint/テスト
- Botによるレビュー補助
(例: Danger, Reviewdog)
- 実行例: PRにコメントで結果通知

品質を「人手＋自動化」で守る。



⚙️ GitHub ActionsでCI/CD

構成例:

```
name: CI Pipeline

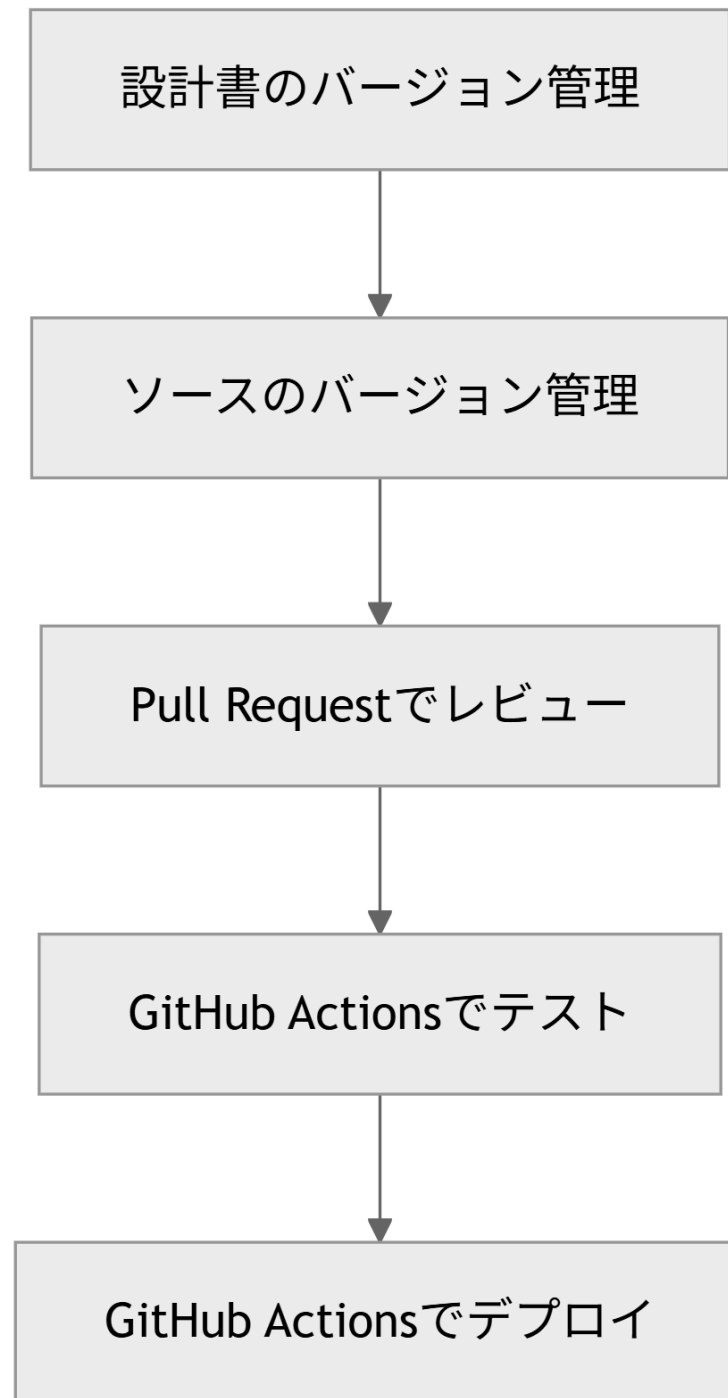
on: [push, pull_request]

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - run: npm install && npm test
```

- ✅ Pushでテスト → デプロイまで自動化
- ✅ 複雑なワークフローもYAMLで定義可能

モダン開発フロー全体像

- 設計・開発・テスト・デプロイの一体化
- すべてをGitHub中心で完結可能！



導入メリットとTips

メリット

- 開発フローの見える化
- 自動化による品質向上と効率化
- ナレッジがGitに集約

Tips

- READMEに設計書へのリンクを明記
- PRテンプレートでレビューポイントを明示
- secretsを使ったセキュアなCD設定

まとめ

- GitHubは開発の中心プラットフォームへ進化
- AsciiDoc × GitHubで設計とコードを一元管理
- ActionsでCI/CDもシームレスに
- チーム開発の質が劇的に向上！

? ご質問はありますか？

 お気軽にどうぞ！

 ご清聴ありがとうございました！

Follow me on GitHub

 <https://github.com/tomotoaki>