

「テストからはじめよ」 ～忍者式テスト20年の実践から～

キヤノンメディカルシステムズ株式会社

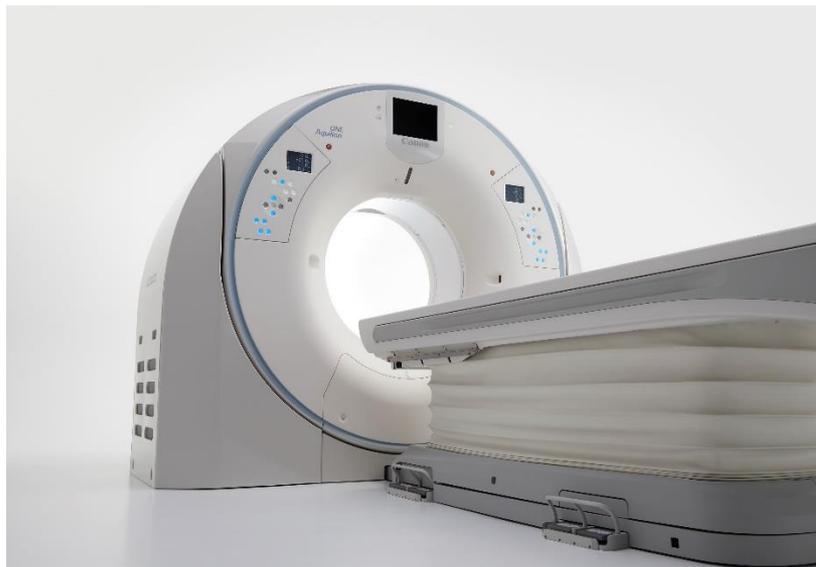
深谷美和

関将俊

13 June 2023

今日の話

- 私たちのチームはX線CT装置をXPで開発しています
- 反復開発の利点と反復開発に有効なプラクティス「忍者式テスト」を説明します



今日の話

➤ 反復開発

- Royceのモデル、V字モデル
- 工程別問題発見能力の違い
- 反復開発の難しさ

➤ 忍者式テスト

- 【図解】忍者式テスト／ストーリー／受け入れテスト／規模と向き合う
- テスト実施の記録（20年分）

➤ 忍者式テストの効果

私たちについて

➤ 深谷美和

- 医療機器（自社製品）のソフトウェア開発に従事
- eXtremeなチームのテスター
- 動かして試すのが好き

➤ 関将俊

- 医療機器（自社製品）のソフトウェア開発に従事
- eXtremeなチームのプログラマ
- Ruby Core Committer

話したくなかったわけ

➤ 変わったチーム

- 「テストを全部手動でやり直してる」と聞いた。 どういうこと？ ？ ？
- 膨大なテストケースを全部？ どうやって？（信じられない）

➤ 2014年、チームにJoinした

- 本当にテストを全部手動でやり直してた！
- いつも「どうやってためすの？」「うまくいったらどうなるの？」から考えている
 - ・ 製品だけでなく、自分たちのあらゆる活動についても、そこからはじめる
- こんなおかしいことを20年も続けているチームは見たことがない
- 紹介したい（書き物として残したい）

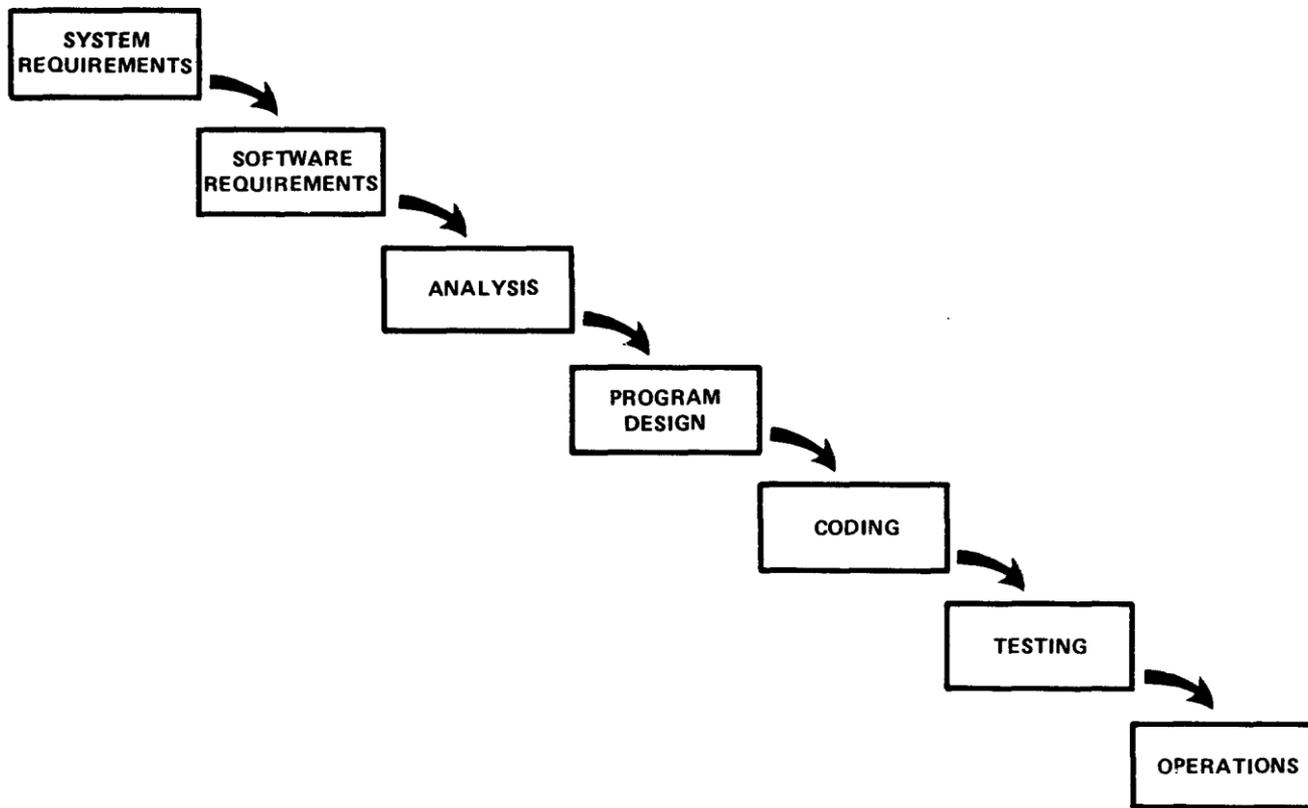


テストからはじめよ

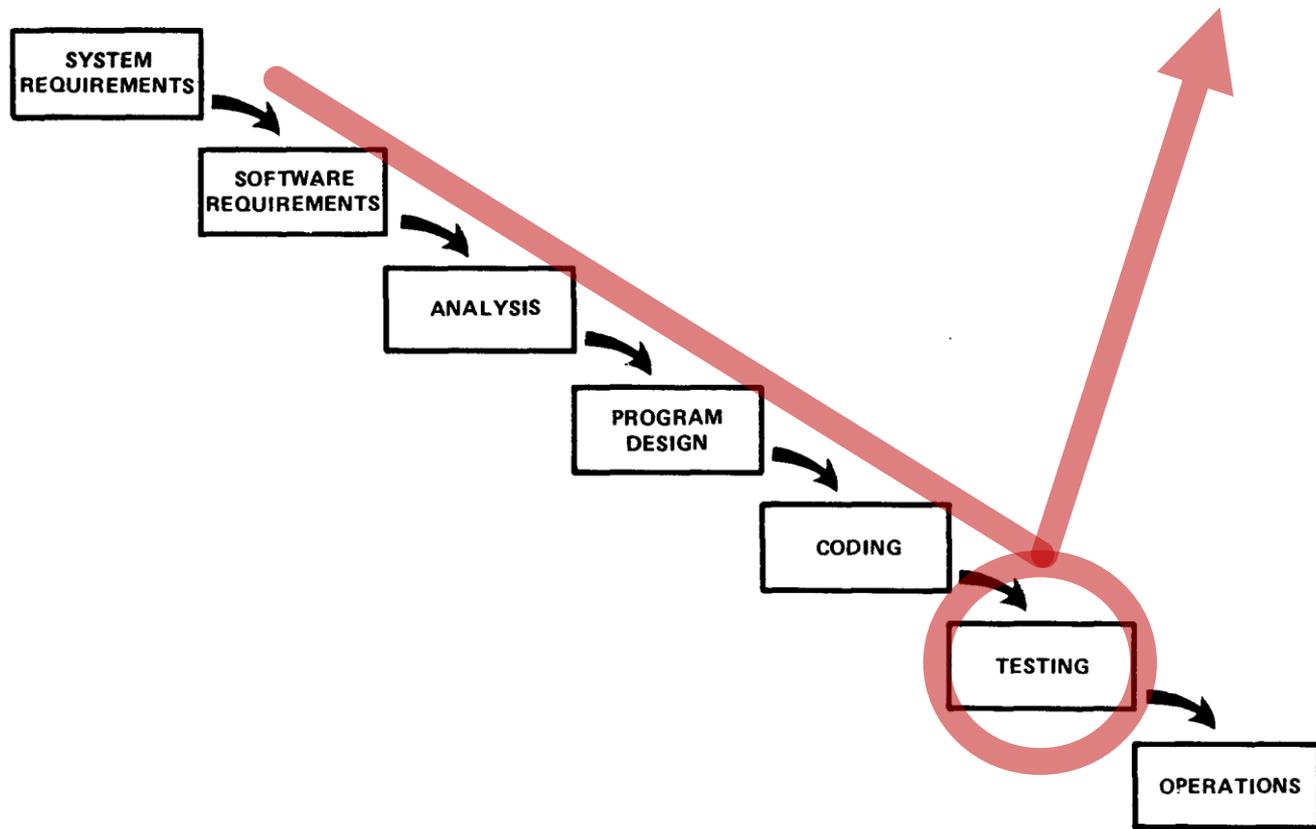
反復開発

- なぜ反復開発なのか
 - Royceのモデル、V字モデル
 - 工程別問題発見能力の違い
 - 反復開発の難しさ

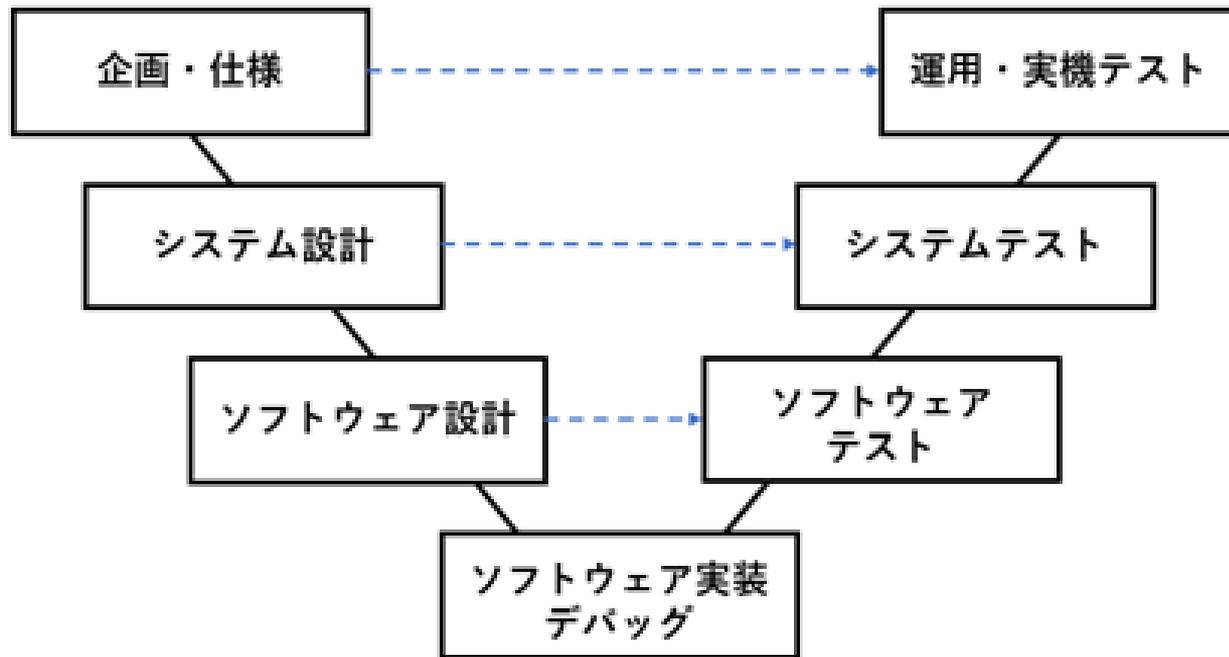
Royceによる大規模ソフトウェア開発のための実行ステップ



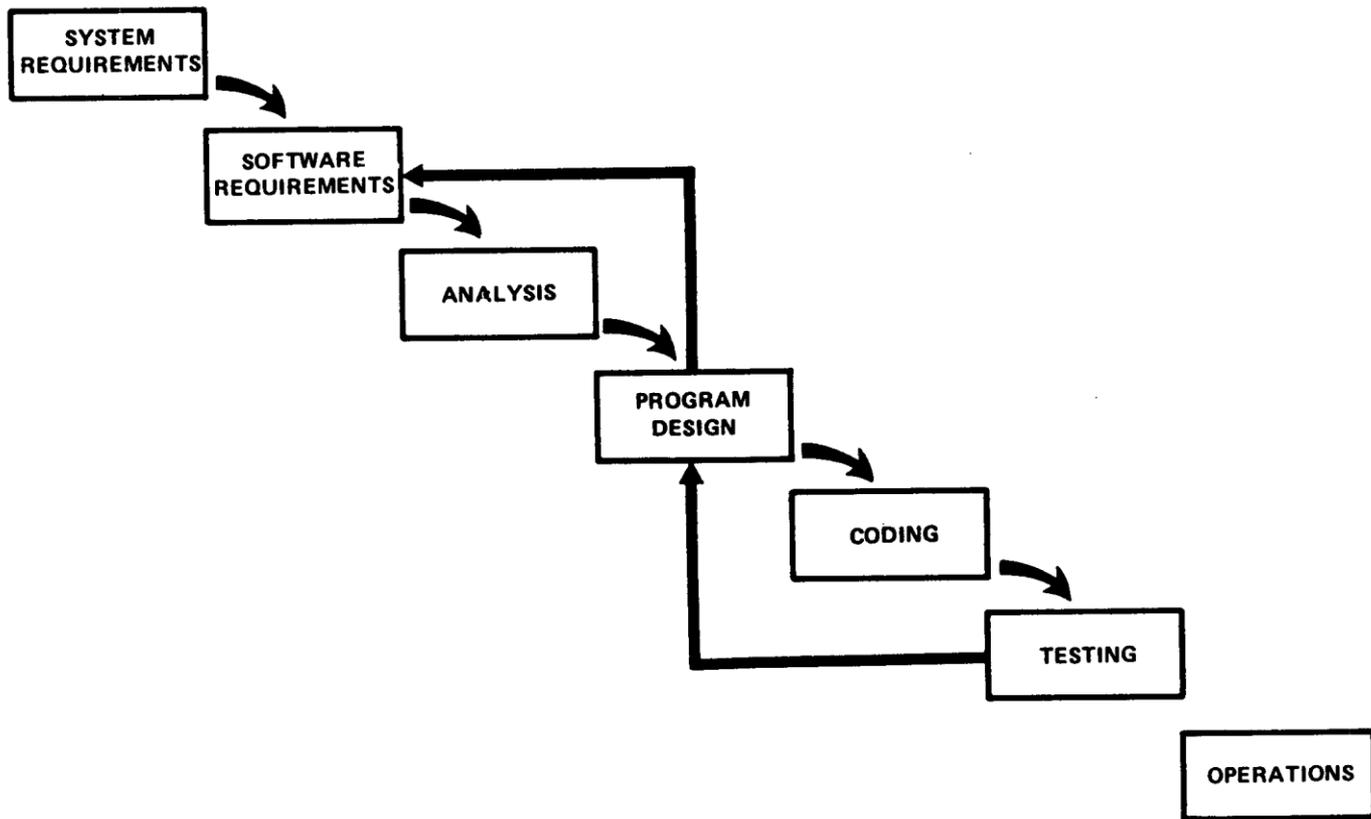
Royceによる大規模ソフトウェア開発のための実行ステップ



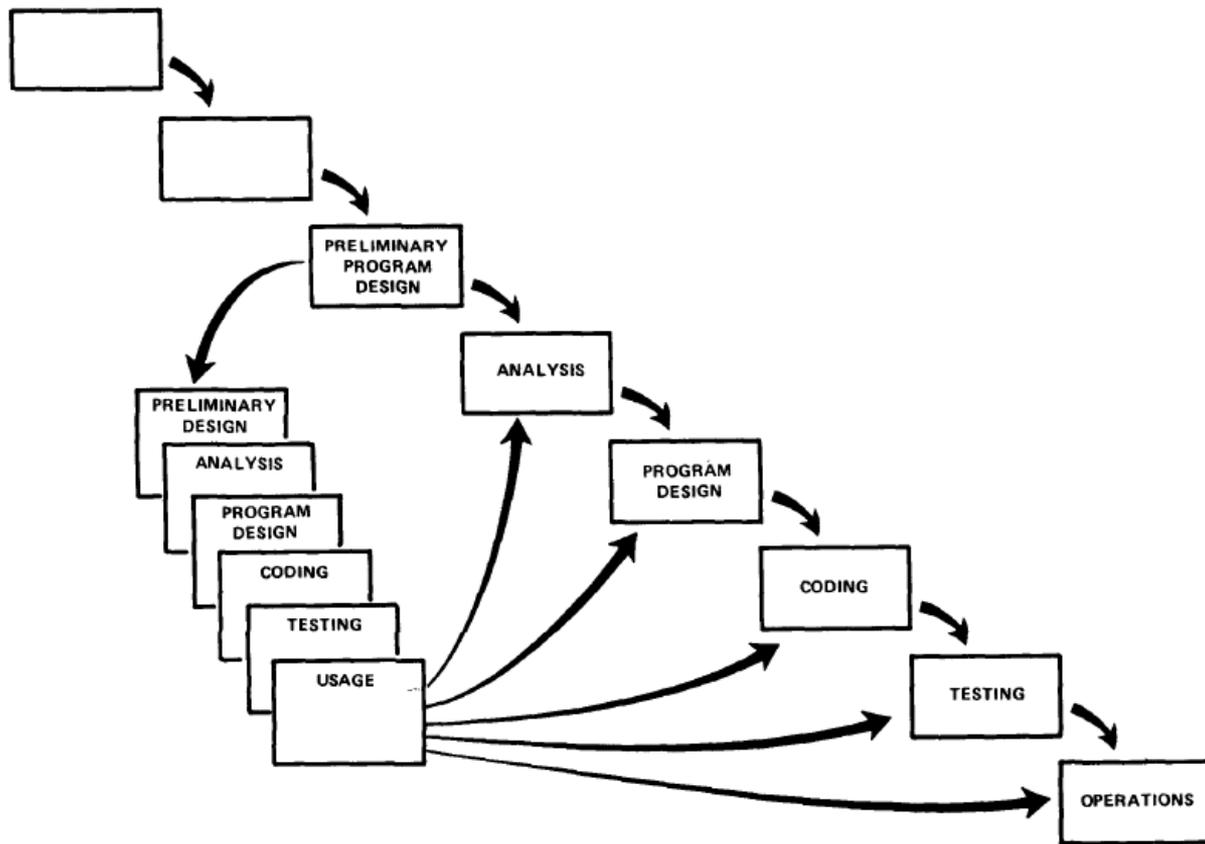
V字モデル



Royceによる大規模ソフトウェア開発のための実行ステップ

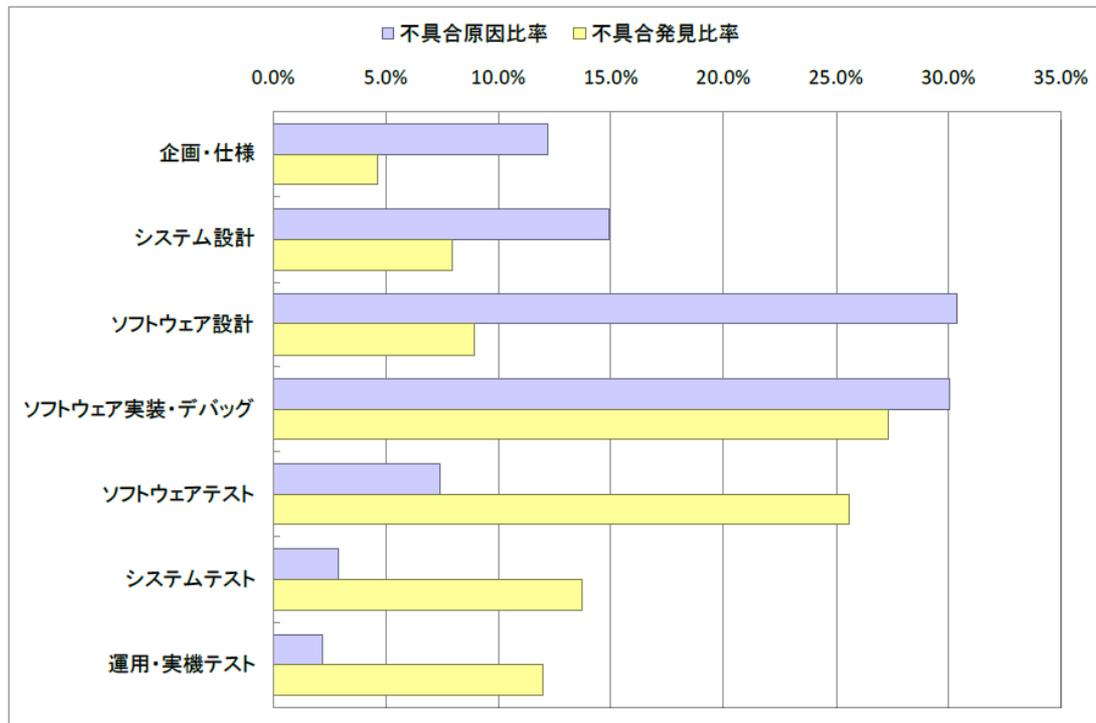


Royceによる大規模ソフトウェア開発のための実行ステップ



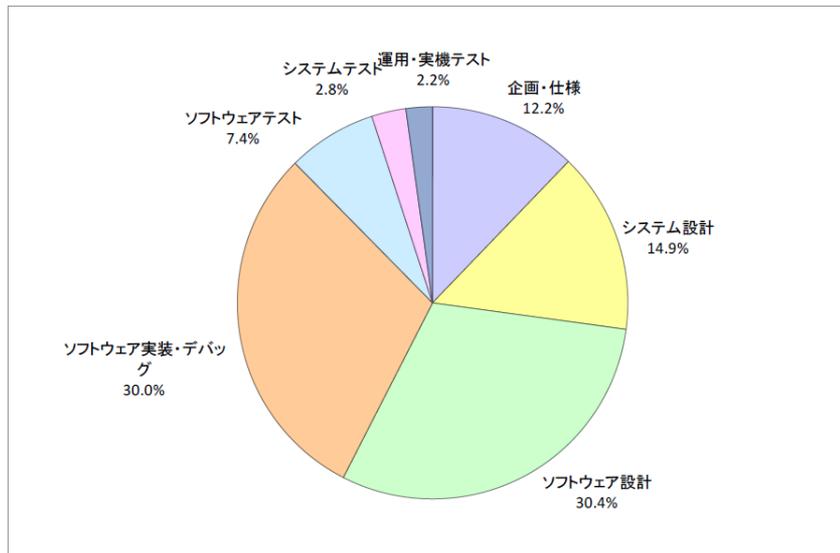
工程別問題発見能力の違い

➤ 不具合の原因工程と不具合の発見工程



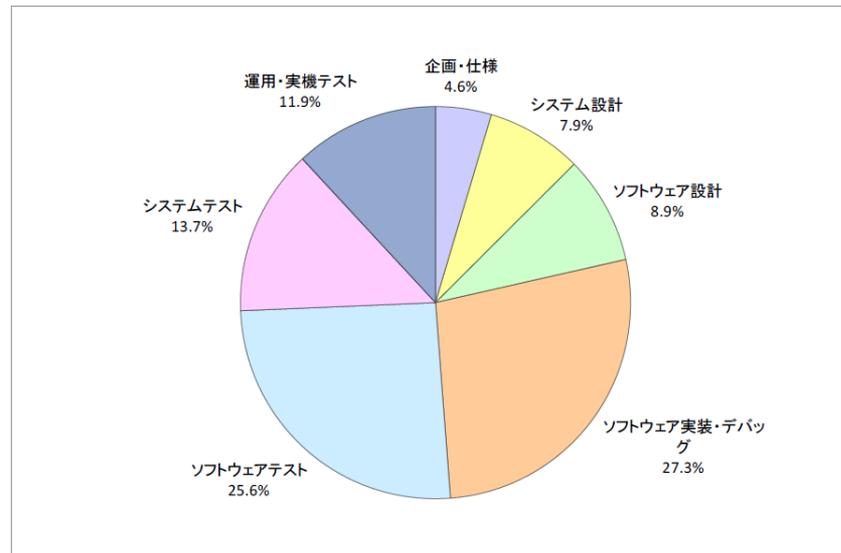
工程別問題発見能力の違い

➤ 不具合の原因工程比率



不具合の原因は「企画・仕様」、「システム設計」「ソフトウェア設計」の上流工程が6割弱となっている。

➤ 不具合の発見工程比率



不具合は「ソフトウェア実装・デバッグ」、「ソフトウェアテスト」「システムテスト」、「運用・実機テスト」の下流工程で約3/4以上が発見されている。

工程別問題発見能力の違い

- 数字からわかる事実
 - 企画・仕様、設計の問題を、その工程で見つけるのは難しい
 - テストは問題を見つける能力が高い

- 試してわかることは試す
 - 試さずに会議やレビューだけですべての問題を発見するのは難しい
 - 中間成果物をあてにするのは危険
 - 試してみたほうが効率が良い
 - ・ ソフトウェアの特徴のひとつ（安く試せる）

工程別問題発見能力の違い

➤ これらを利用する

- 工程と時期を分けて考える（工程≠時期）
- 問題を発見する能力が高い工程をプロジェクトの初期から行う

- 早い時期に問題を発見できる可能性が高まる
- 早い時期に問題を発見できれば、修正が容易になる

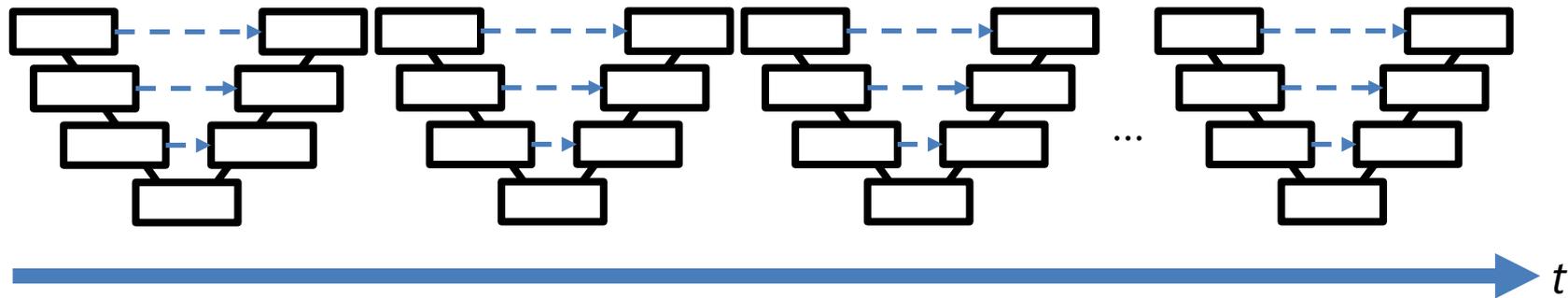
➤ 作戦

- システムテスト、運用・実機テストをより早い時期に行う
- **結合された状態でのテストを早く、頻繁に！**

そこで反復開発ですよ

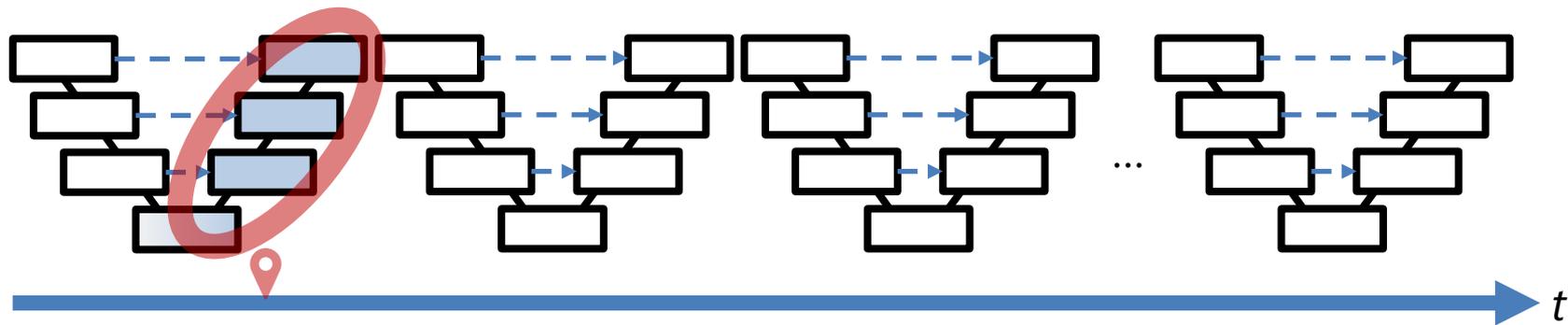
そこで反復開発ですよ

- テストを早い時期から実施するため、プロジェクト全体をn回のV字で構成する



そこで反復開発ですよ

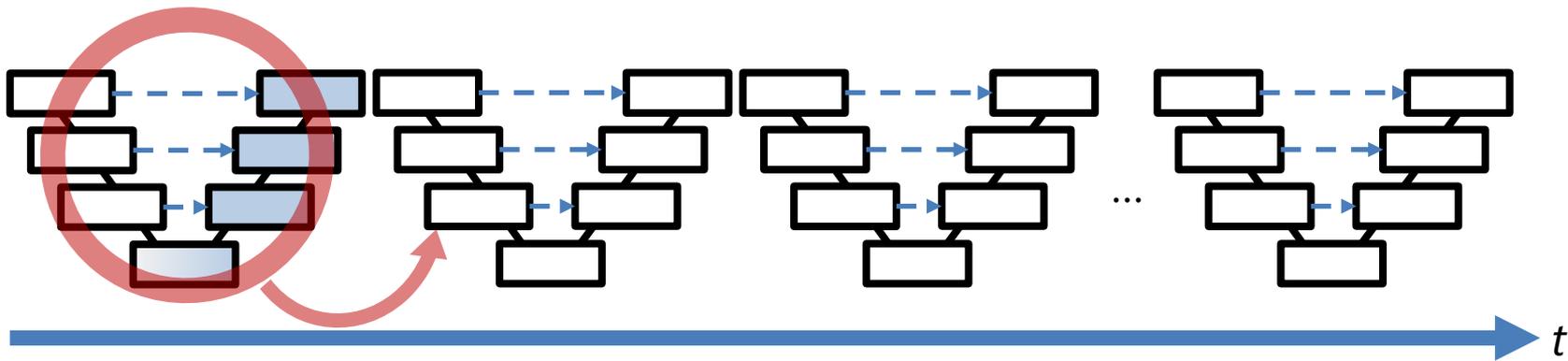
- テストを早い時期から実施するため、プロジェクト全体をn回のV字で構成する



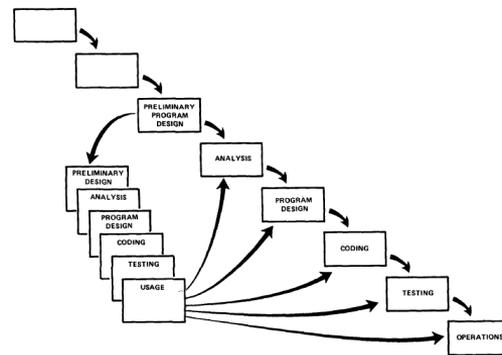
- 初回のV字（プロジェクトの初期）から、テストが実行される

そこで反復開発ですよ

- 次のV字に前回のV字で発見した問題や違和感、知見をフィードバックできる

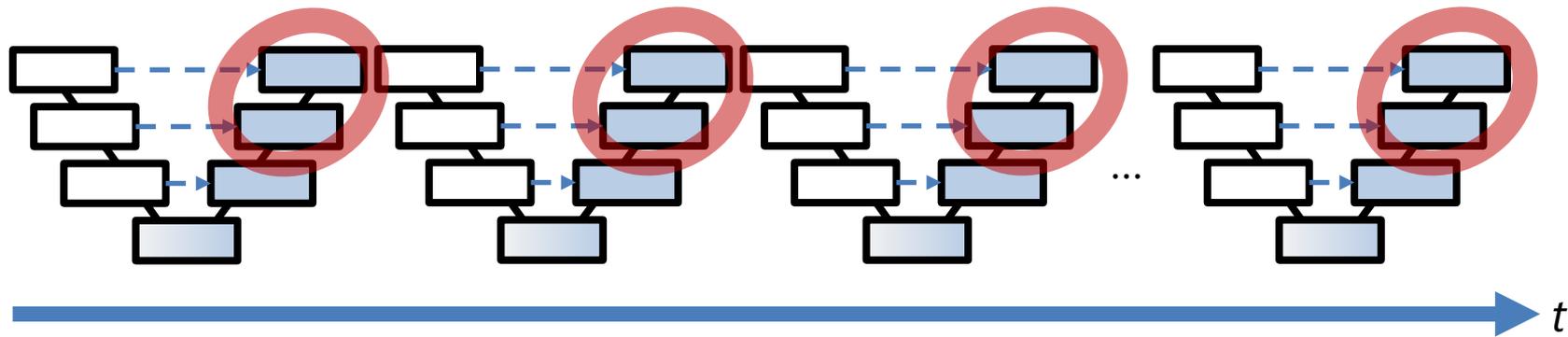


- Royceの主張は2回だけど、n回やってもいい



反復開発の難しさ

- 頻繁にシステムテスト、運用・実機テストを実施しなければならない



- 新しい反復により改定された仕様の確認
- それまでに搭載したものが全て問題なく動作することの確認

そこで忍者式テストですよ

そこで忍者式テストですよ

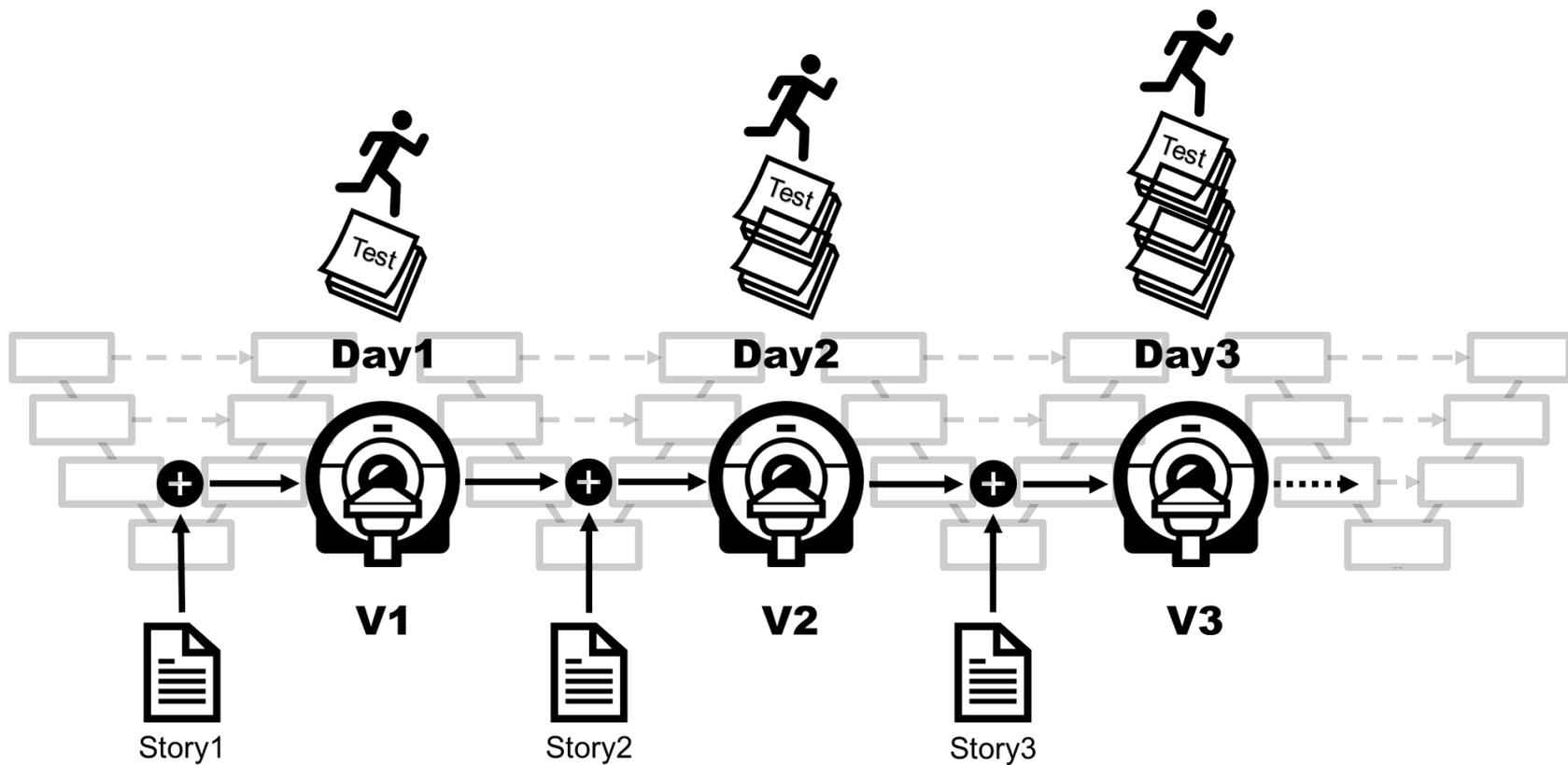
➤ 忍者式テスト

- 【図解】忍者式テスト
- ストーリー
- 受け入れテスト
- 規模と向き合う
- テスト実施の記録（20年分）

そこで忍者式テストですよ

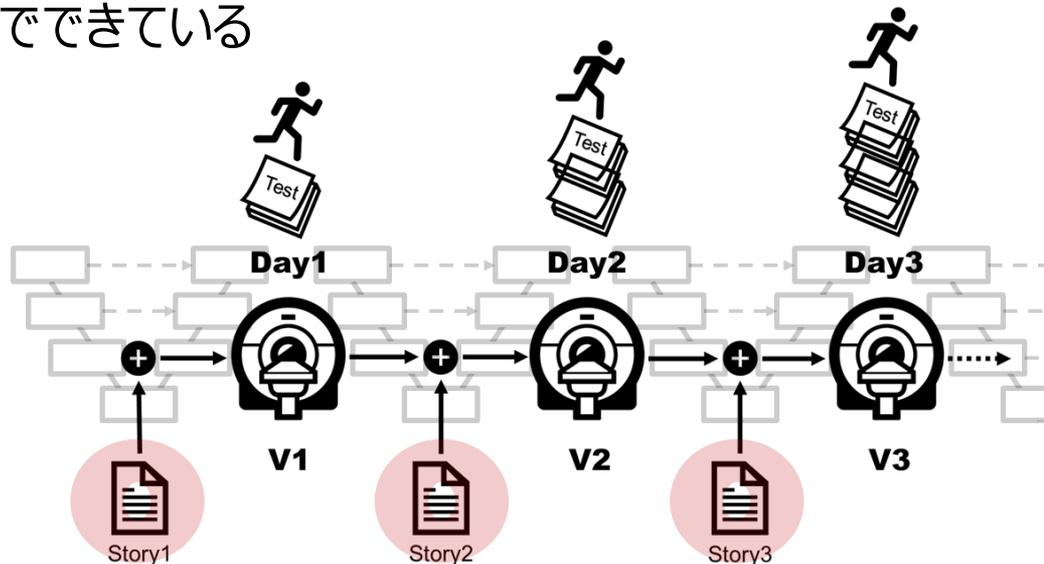
- テスト駆動開発を受け入れテストのレベルで行う作戦
 - xUnitを使ったTDDのように、受け入れテストからはじめる開発
 - テストケースに導かれてプログラムを開発するように、受け入れテストに導かれてストーリーを実現する
 - ストーリーが増えるたびに、それまでに作った全てのストーリーの受け入れテストをやり直す

【図解】忍者式テスト



ストーリー

- ストーリーはXPなどの反復開発で製品を変更する単位
 - ユーザーにとってシステムがどのように変化するか
 - 受け入れテストで確認する
- 製品はたくさんのストーリーの集まりでできている



ストーリー



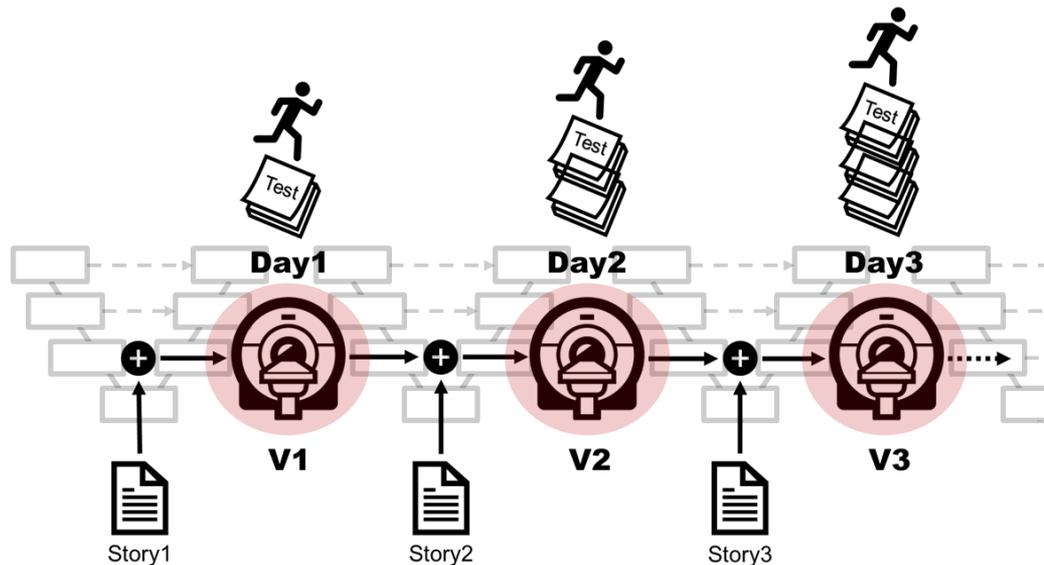
Story

- チケット
 - 概要、要求の背景
 - テストケース
 - ・ システムの具体的な変化と確認方法
 - 開発日記
 - ・ 毎日の試行錯誤の様子、実験した内容や結果 など
 - ・ 設計や実装の根拠がわかる

- ストーリーは数日で完成する大きさ
- 開発中に見つけたバグもストーリーと同様に扱う

受け入れテスト

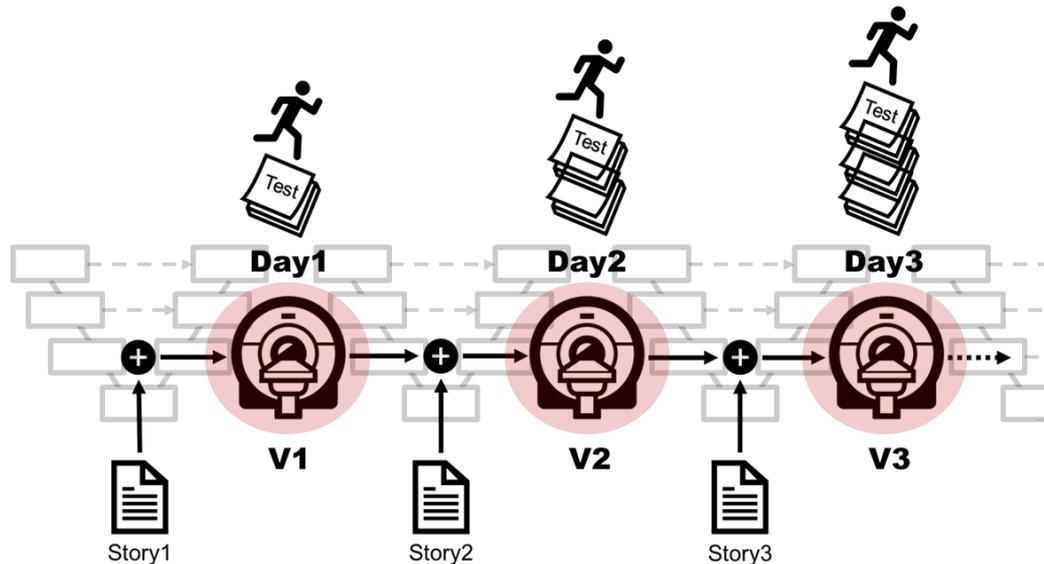
- システム全体で結合した状態で試す
- ユーザーが使うのと同じように一連の流れで試す



なぜ手動でテストするのか

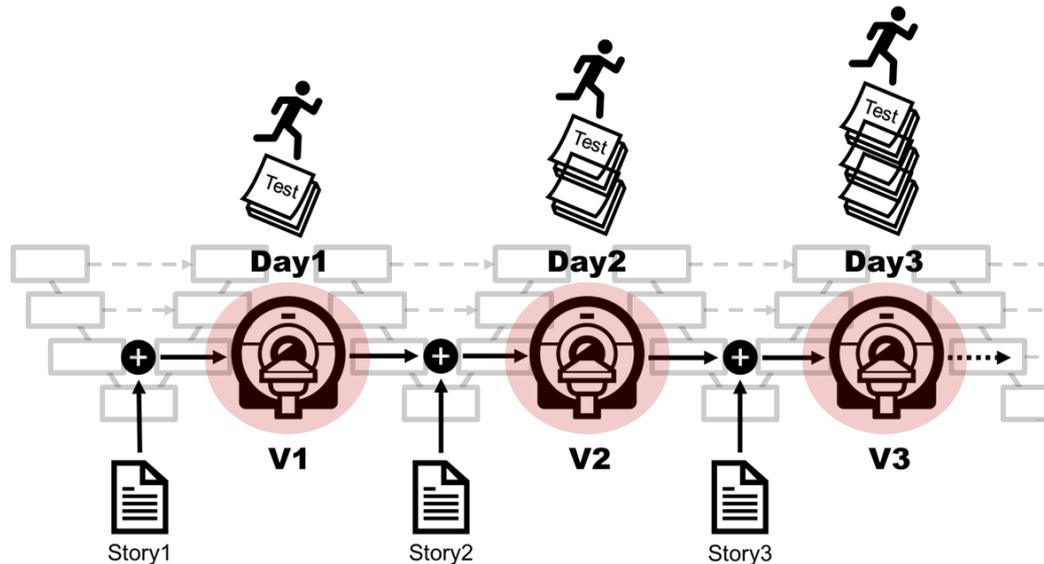
- 私たちが見つけたいもの → 理想との差分
 - 使いづらさや誤解をまねく仕様
 - 手触りの良し悪し
 - ・ 反応の自然さ、再描画の様子、チラつき、なめらかさ など
 - テストケースの誤り
 - 開発のやり方自体の問題

- 手動テストが適している

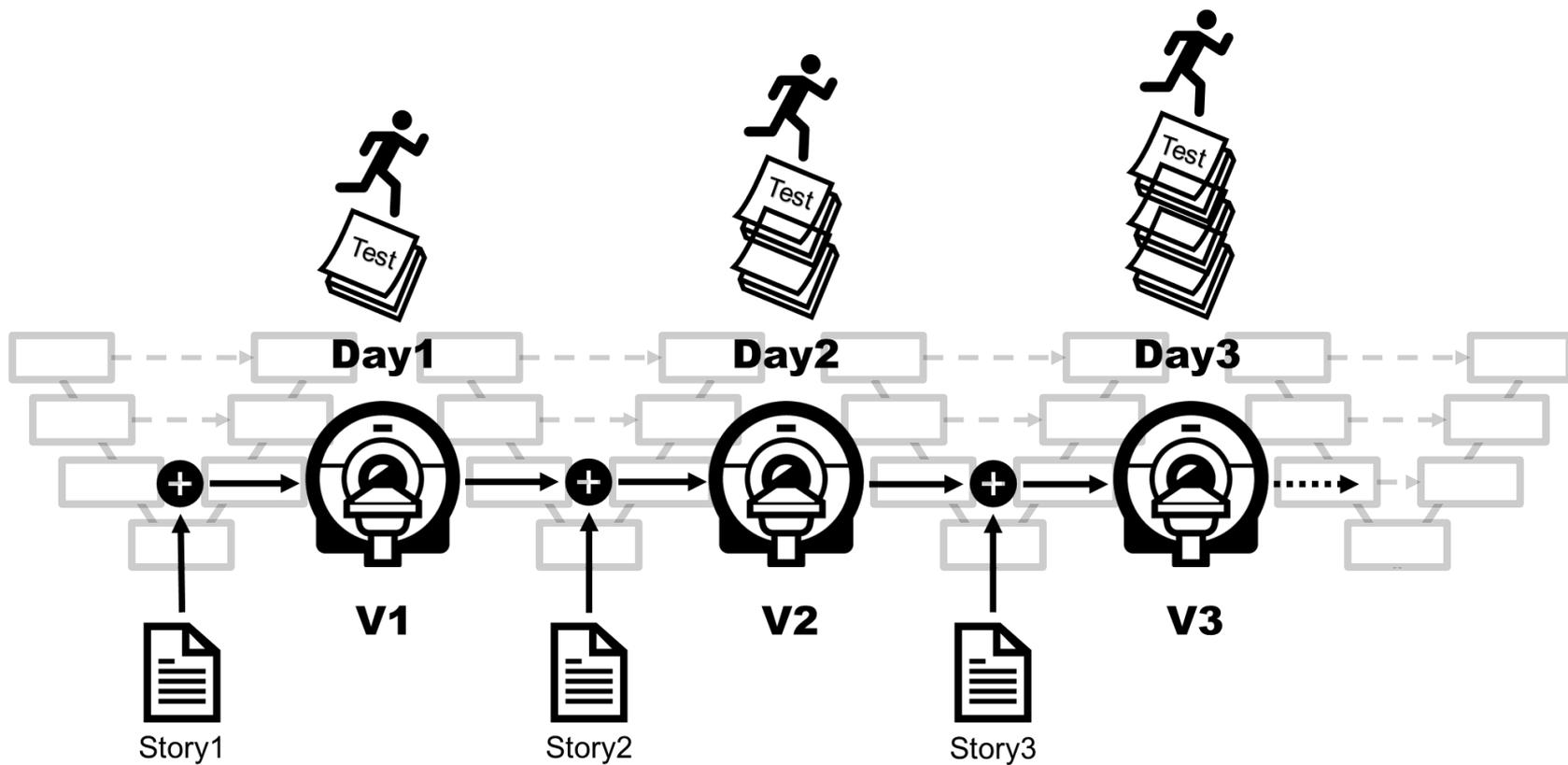


なぜ手動でテストするのか

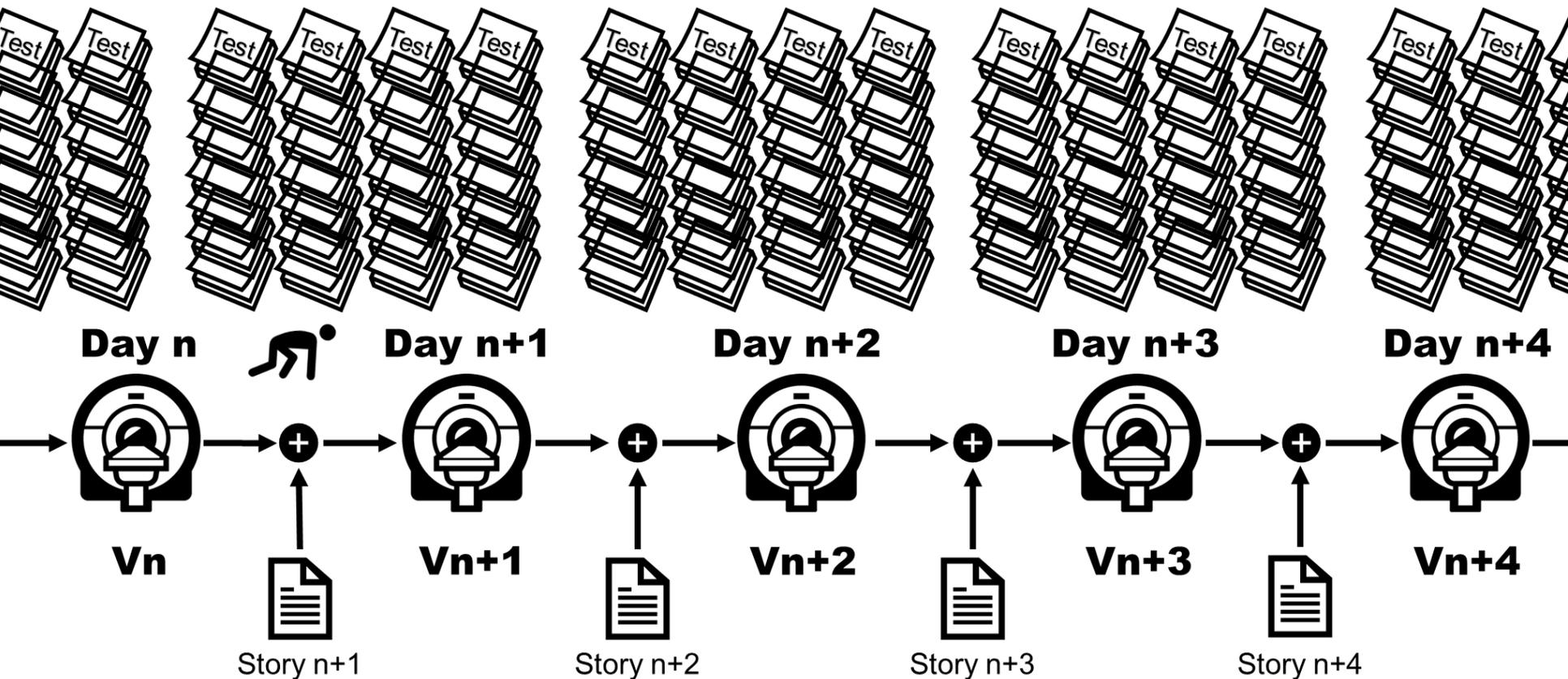
- 理想もアップデートする
 - 製品に求めるレベル、よい製品のイメージ
 - ・ 市場や環境の変化
 - ・ 自分自身の理解
- 手動テストが適している



【図解】忍者式テスト



たいへんなことになります

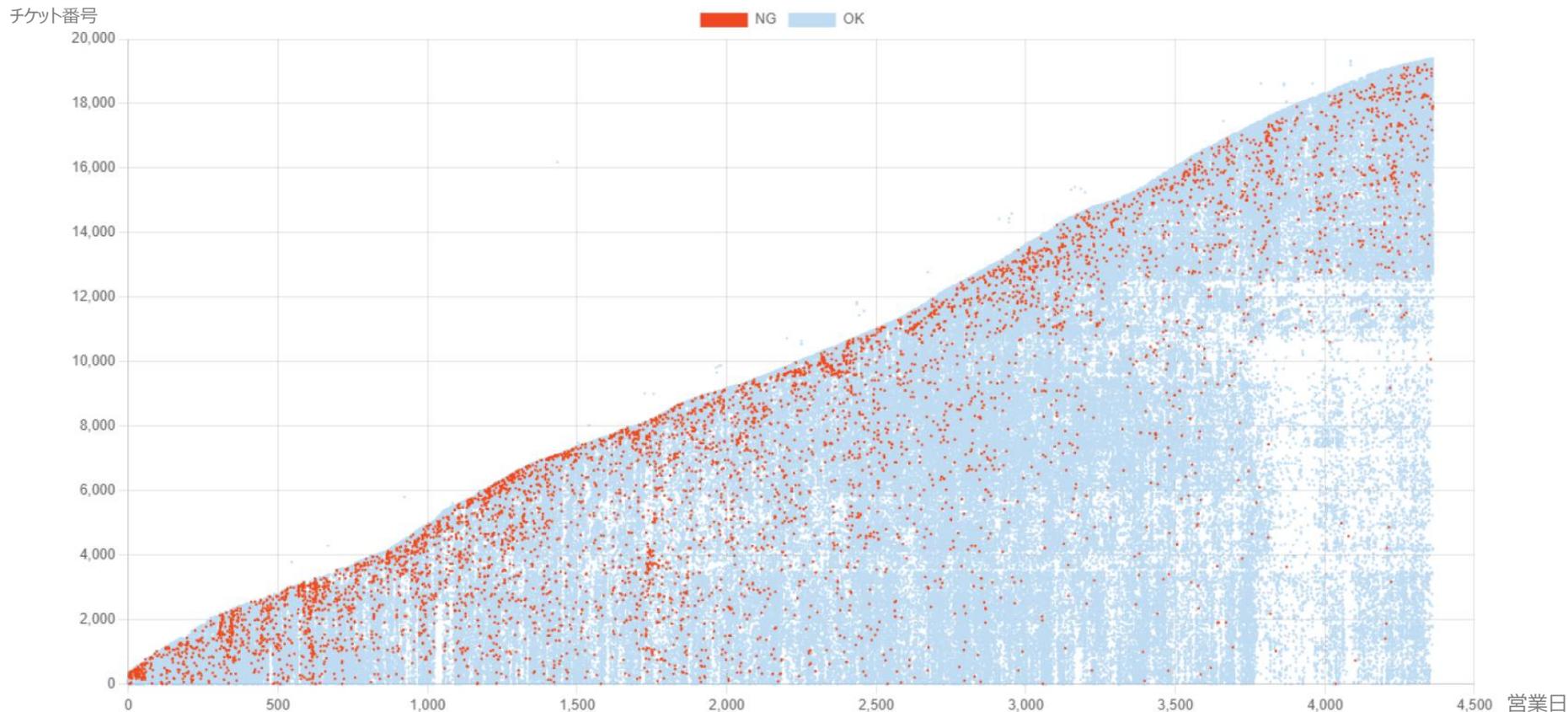


規模と向き合う

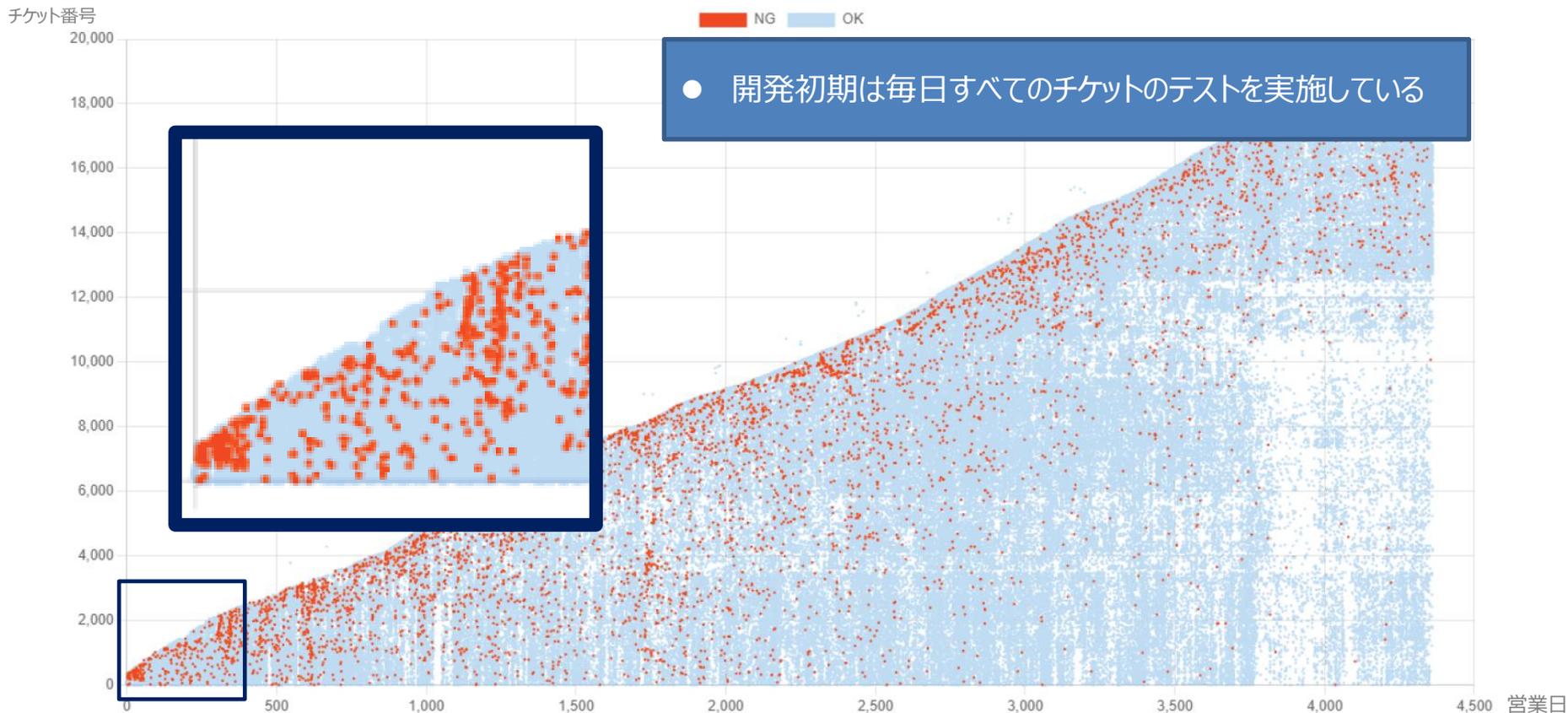
規模と向き合う

- 一定期間でテストケースをすべて回す作戦に切り替えた
- まんべんなく、かつ、効果の高いテストケースを抽出するアルゴリズムの開発
 - 新しいストーリー、修正したばかりのチケットのテストケースを最優先
 - 前回のテスト結果がパスしたテストケースの出現頻度を徐々に落とす
 - 開発の状況に応じて機能ごとに出現頻度を調整
 - ある期間で見ると、すべてのテストケースが実行できる
- アルゴリズムにより抽出した今日のテストスイート → 「本日のおすすめテスト」
 - 一日にできそうな量のテストケースしか表示しない（量に圧倒されないようにする）

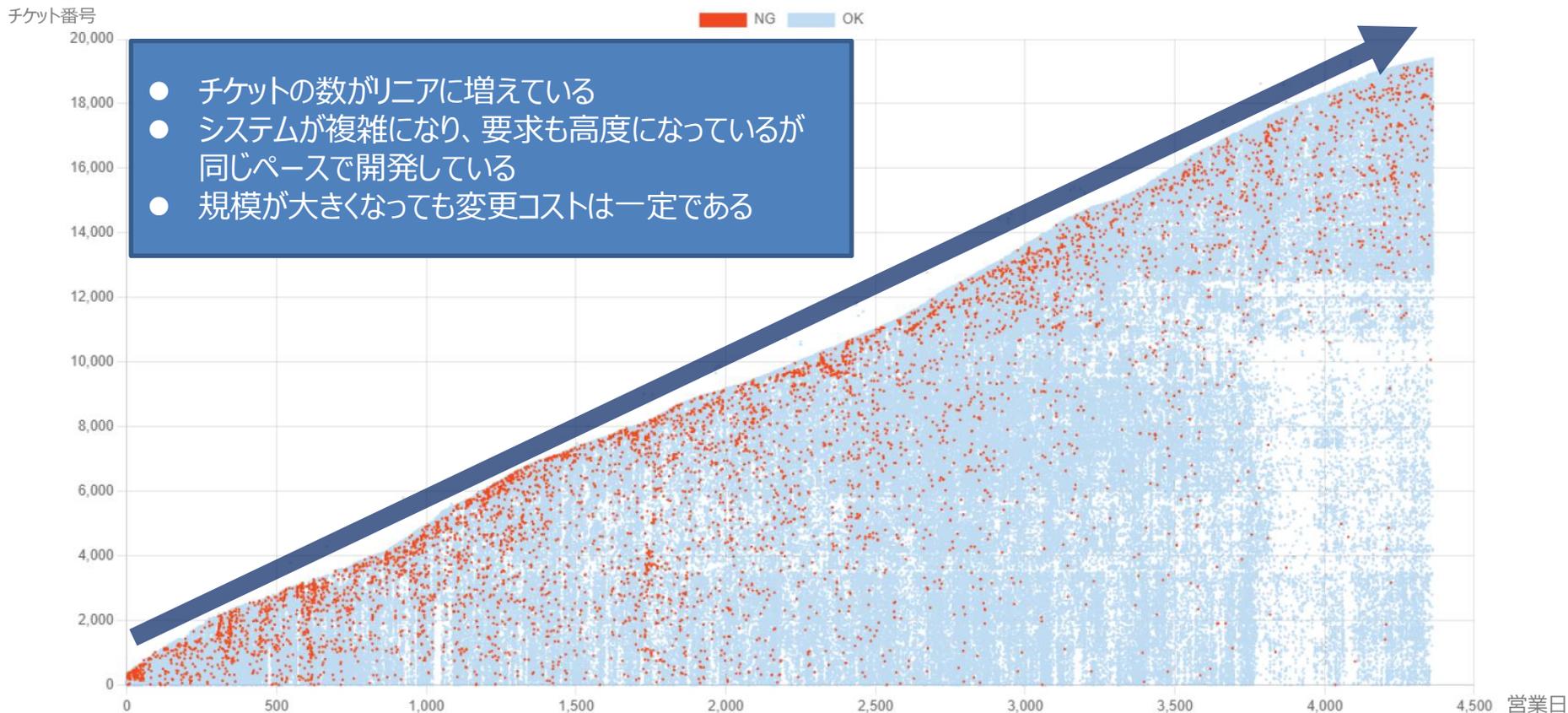
テスト実施の記録（20年分）



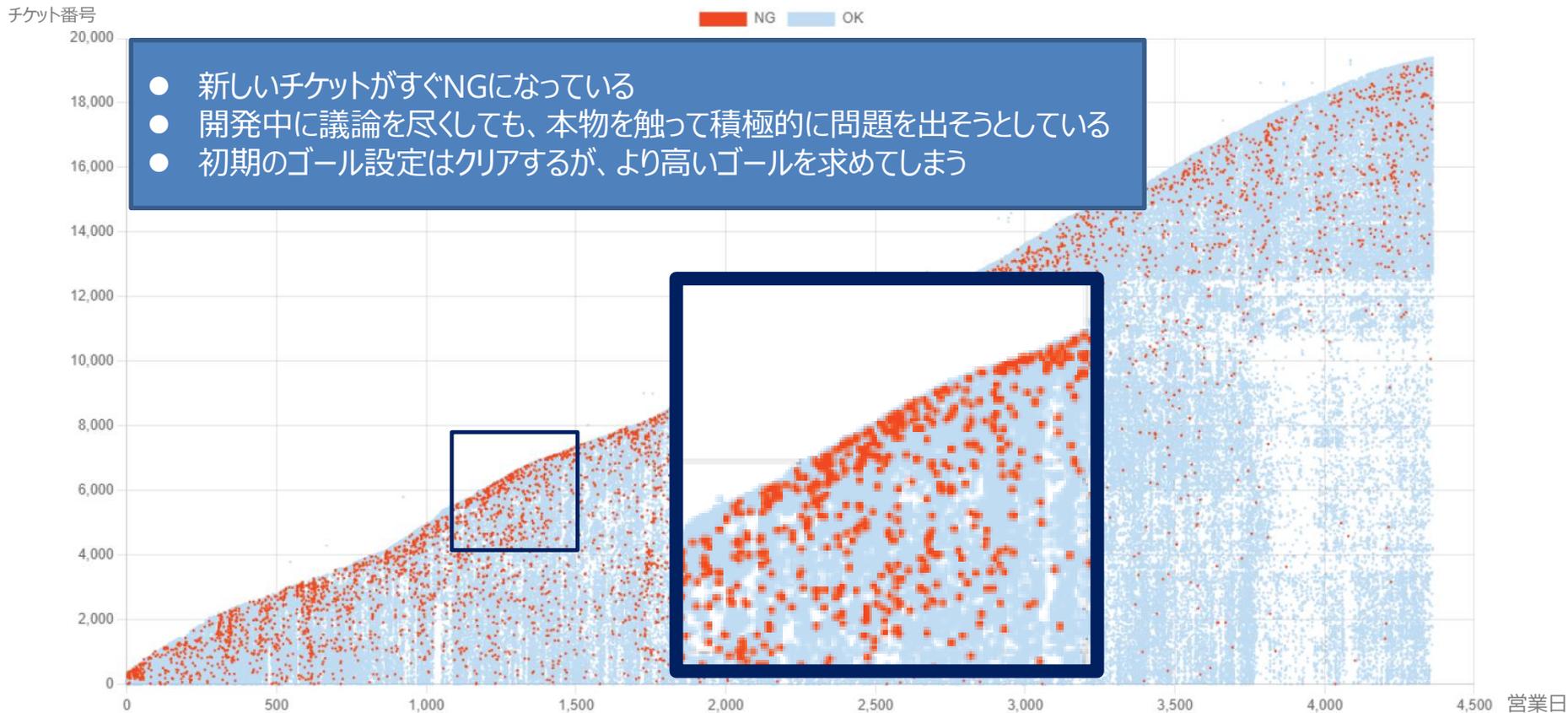
テスト実施の記録（20年分）



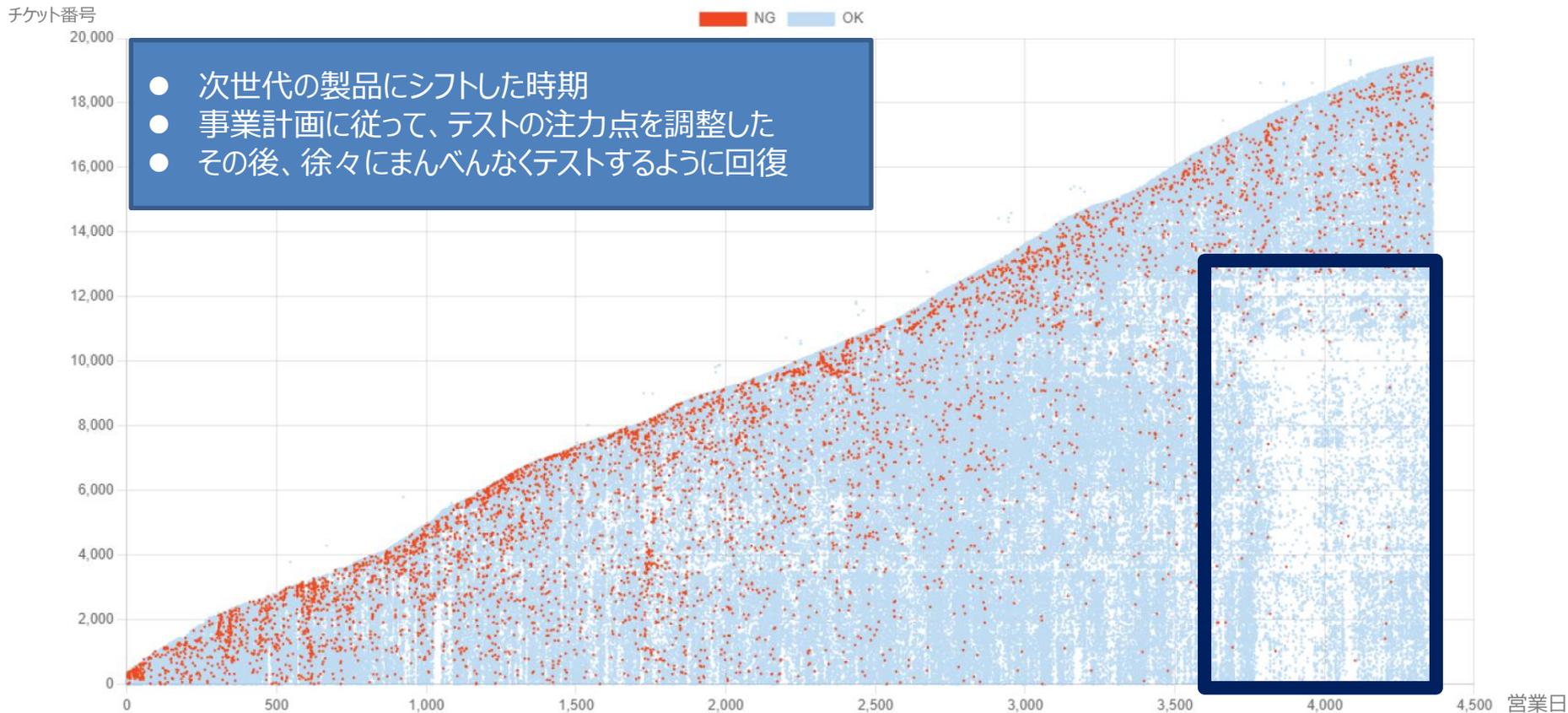
テスト実施の記録 (20年分)



テスト実施の記録（20年分）



テスト実施の記録 (20年分)



忍者式テストの効果

- 忍者式テストの効果
- 副次的な効果

忍者式テストの効果

- 難しそうな問題も躊躇せずに修正できるようになった
- 不具合や性能などの実装レベルの問題に積極的に対応できるようになった
- 直せる幅が広がった
- 理想の製品をイメージして、それとの差分も積極的に探しだす者が現れた
- 全員がずっと「良い製品とは何か」を問いながら開発できるようになった

副次的な効果

➤ 学習機会の創出

毎日のテスト時間、朝会のチケットの読み合わせなどさまざまな場面で、製品の仕様、設計、実装など、製品に関する質疑応答が絶えず行われる。これは圧倒的な量のドリル学習に相当する。これにより、ひとりひとりの脳内にシステムのあらゆることが配線され、製品を動かしたときに瞬時におかしさに気づくようなことが起きる。誰かのちょっとした疑問が別の誰かの思考を刺激する。チームはひとつの巨大な脳を持った生き物のようになる。

今日の話

- 反復開発
 - Royceのモデル、V字モデル
 - 工程別問題発見能力の違い
 - 反復開発の難しさ

- 忍者式テスト
 - 【図解】忍者式テスト／ストーリー／受け入れテスト／規模と向き合う
 - テスト実施の記録（20年分）

- 忍者式テストの効果

Made For life

患者さんのために、あなたのために、そして、ともに歩むために。

人々の健やかな生活の実現のために、「いのち」と向き合う。

「Made for Life」はキヤノンメディカルシステムズの経営理念を象徴するスローガンです。

Canon

キヤノンメディカルシステムズ株式会社