

「生成AIで自動化」だけでテストは十分？ 自動・手動の最適バランス

2024年 9月 18日

株式会社GENZ

IT技術部 技術推進Group

竹花 和裕

会社紹介

会社名	株式会社GENZ	
拠点	東京本社	〒101-0062 東京都千代田区神田駿河台2-3-11 ヒューリック御茶ノ水ビル3F
	札幌支店	〒060-0061 北海道札幌市中央区南1条西5-20 郵政福祉札幌第1ビル6F
	青森支店	〒030-0861 青森県青森市長島2-25-3 ニッセイ青森センタービル6F
事業内容	品質コンサルティング/システム開発/システム技術者派遣 損害保険代理店業/不動産売買・賃貸仲介業務 営業・マーケティング支援	
設立	2016年5月	
従業員	240人（2024年9月時点）	

GENZのテスト



QUALITY ENGINEERING

品質を考えるサービスカテゴリとして下記支援内容を提供しております

- ・ 品質コンサルティング
- ・ QMOサービス
- ・ テスト計画/設計
- ・ ドキュメント作成サービス



TEST EXECUTION

WEB・スマホアプリをはじめとしたテスト実施の支援をおこないます

- ・ PCテスト
- ・ スマートフォンテスト
- ・ 各種組込み機器テスト
- ・ ゲームデバッグ
- ・ IoTデバイステスト



Non-Functional Requirements

脆弱性テスト、ストレステスト、自動化テスト等の特定のツールが必要なテストを支援します

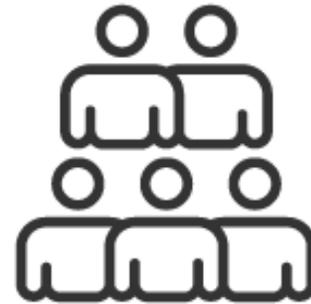
- ・ 脆弱性診断
- ・ 負荷テスト
- ・ テストツール作成

GENZの強み



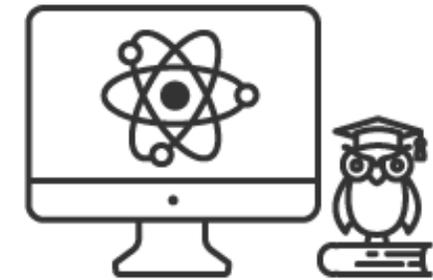
多様な経験による柔軟な対応

徹底した品質管理を行うためエンジニアのスキル向上や多様な経験ができるようマネジメントしています。そのため、お客さまのご予算・ご要望に応じて高品質かつ最適なテストサービスを提供することができる体制を整えています。



人日単位での発注可

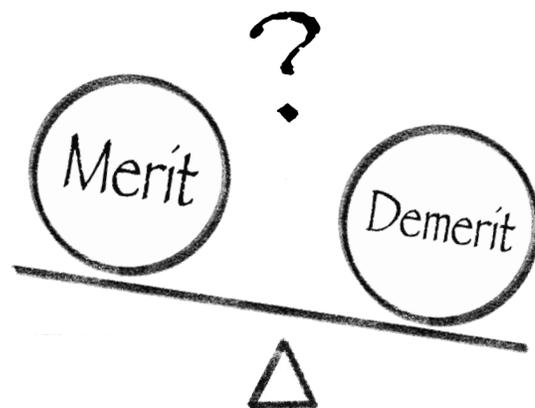
SI業界では「人月」単位でのテスト業務請負が多い中、GENZではお客さまの様々なニーズに対応するために、「人日」単位でのご依頼も可能です。お客さまのニーズに柔軟かつスピーディに対応しております。



エンジニアの技術ノウハウが蓄積

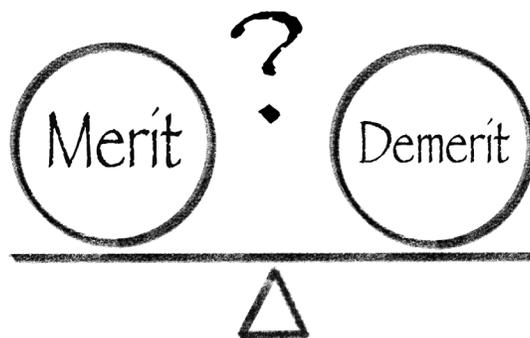
当社2022年度の離職率は5.07%です。GENZで長く活躍している経験豊富なメンバーが、お客さまに寄り添って伴走いたします。

「生成AIで自動化」だけでテストは十分？
自動・手動の最適バランス



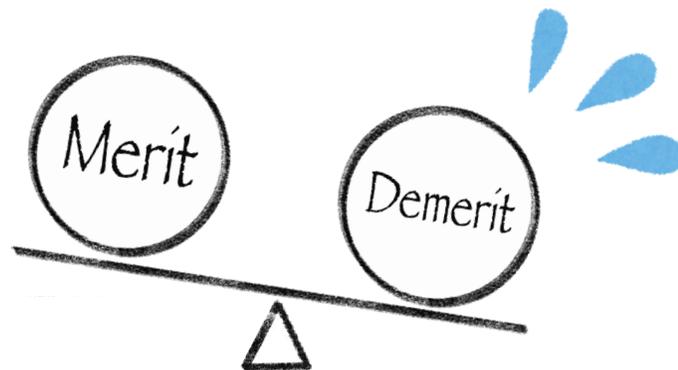
結論ファースト

自動・手動の最適バランスは



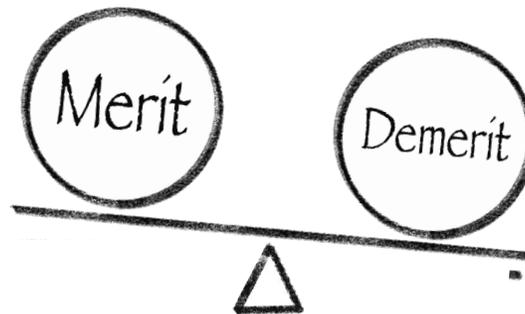
結論ファースト

自動・手動の最適バランスは **ありません**



結論ファースト

自動・手動の最適バランスは **ありません**



最適バランスを取るためにおさえるポイントがあります

結論ファースト

自動化を検討した際に分岐点ごとにお話しします

- 単体テスト vs システムテスト
- 手動テスト vs 自動テスト
- 人 vs AI



自動テストをどこでやる？

単体テスト vs システムテスト

単体テスト vs システムテスト

テスト駆動開発の考え方や開発フレームワークの進化で
単体テストの自動化が進んでいる

しかし、**E2Eのシステムテスト**はまだまだ**手動実施も多い**



手動のシステムテストは**コストが大きい**ので自動化したい

単体テスト vs システムテスト

しかし、自動化したシステムテストもコストがかかる

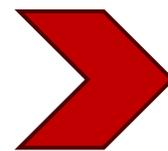
- UIを操作するので**テストの時間**がかかる
- ネットワークの影響で**テストが安定しない**
- バグの可能性が複数ある → **デバッグに時間がかかる**
 - システムが複雑
 - テストの不安定性
 - テストコード
- UIの変更に弱い
- **自動テスト自体の構築が必要**

単体テスト vs システムテスト

しかし、自動化したシステムテストもコストがかかる

- UIを操作するのでテストの時間がかかる
- ネットワークの影響でテストが安定しない
- バグの可能性が複数ある → デバッグに時間がかかる
 - システムが複雑
 - テストの不安定性
 - テストコード

- UIの変更に弱い
- 自動テスト自体の構築が必要



ローコードや**ノーコード**で
自動テストを作れるサービス
でフォローも可能

単体テスト vs システムテスト

システムテスト自動化は**3回以上同じテストを繰り返す際**に有効

• 自動テスト**構築コスト**や**メンテナンスコスト**で赤字になる

自動テストは**なるべく単体テスト**で行うのが望ましい

システムテストの自動化は**システムテストでなければならない箇所**に絞る

ところで、**単体テストってどこまで？** 結合？ システムって？

テストレベルではなく**テストサイズ**で考えるのがオススメ

単体テスト vs システムテスト

テストサイズ：Google社内から広まった自動テストに対する分類

- Small
 - **単一のプロセス**で実行
 - ファイルシステムやDBの**外部リソースは使わない**
 - テスト実行時間は**1分以内**（目標は1ミリ秒未満）
- Medium
 - **ローカル環境**で実行
 - **コンテナ利用**なら外部リソースを使ってもOK
 - テスト実行時間は**5分以内**（目標は1秒未満）
- Large

単体テスト vs システムテスト

テストサイズ：Google社内から広まった自動テストに対する分類

- Small

- 単一のプロセスで実行
 - ファイルシステムやDBの外部リソースを使わず
- テスト実行時間は1分以内（目標は1分未満）

- テスト時間が短い
- 手軽
- 本番環境に遠い

- Medium

- ローカル環境で実行
- コンテナ利用なら外部リソースを使わず
- テスト実行時間は5分以内（目標は1分未満）

- テスト時間が長い
- 手間がかかる
- 本番環境に近い

- Large

【まとめ】単体テスト vs システムテスト

なるべく単体テスト（Small）を自動化するのが理想

- テスト実行が**安定する**
- コーディング後即テストができ、**すぐ結果が分かる**ので安心

ただ

- **レガシーシステム**で単体テストが組み込めない
- 組み込んだらバグるリスクがある
- **B2Bシステム**でUIの変更がほぼない、UXは優先度が低い
→E2Eの自動テストの**メンテナンスの可能性が低い**
- そもそも**E2Eでテストしないと意味がない**

全自動にお任せ？

手動テスト vs 自動テスト

手動テスト vs 自動テスト

自動テストの利点

手動テスト	自動テスト
疲れちゃう	疲れ ない
たまにやる気が...	やる気 が迷子にならない
もちろん休日は休むよ	夜中 や 休日 もテスト実施可能
たまにテストミスも...	指示通り テスト実施可能

手動テスト vs 自動テスト

自動テストの欠点

- **指示通り**テストを実行してくれる
 - **指示がないと**テストができない
 - テストの**仕様**や**期待値**
 - テストケースの更新漏れなどへの**柔軟性がない**
- 不具合一つでテストが**止まってしまふ**（コードで対応できるが）
- **テスト箇所しか**見ない
- テストに対して**疑問を持たない**（目的、優先度）
- 自動テストでは**検出できない不具合**がある
 - スクロールのカクツキ、ユーザビリティ（使いやすさ）

【まとめ】手動テスト vs 自動テスト

- 自動テストに全振りでは**漏れるテスト**がある、**柔軟性も低い**
 - **人の目での判断**が必要な場所は必ずある
- 自動テストの**構築にコスト**がかかる
 - **自動テスト自体に不具合が混入**する可能性もある

じゃあ、**手動で全部がんばる！**は良くない。木こりのジレンマ

- どこを手動でテストするか、自動でテストするかを**事前に決め効率を上げるため**の自動化であることを忘れない
- **小さなテスト**の自動化でも役にたつ
 - 慣れないうちは**ミニマム**に

AI様よろしく！

人 vs AI

人 vs AI

最近のAIの進化がすごい

- Open AI の ChatGPT
- Anthropic の Claude
- Google の Gemini
- Apple の Apple Intelligence

- Meta の Llama
- Google の Gemma
- Microsoft の Phi-3



人 vs AI

生成AIで、システムテストの自動テストのコード



添付画像のWebサイトの自動テストをPlaywrightとPythonで書いて。

```
python 📄 コードをコピーする  
  
from playwright.sync_api import sync_playwright  
  
def run(playwright):  
    browser = playwright.chromium.launch(headless=False)  
    context = browser.new_context()  
  
    # ページを開く  
    page = context.new_page()  
    page.goto("URL_OF_YOUR_WEBSITE")  
  
    # 自動化のテストケース  
    # 例えば、特定の要素が存在するかを確認するテスト  
    page.wait_for_selector("text=GENZの自動化ソリューション")  
  
    # 他の要素をチェックする  
    page.wait_for_selector("text=ツール導入だけで最適な自動化ができますか?")  
  
    # ボタンをクリックして、次のページに遷移するか確認  
    page.click("text=お問い合わせ ご相談・資料請求")  
    page.wait_for_load_state('networkidle')  
  
    # さらにテストを追加  
  
    # ブラウザを閉じる  
    browser.close()  
  
with sync_playwright() as playwright:  
    run(playwright)
```

雑な指示でもテストを書いてくれる!

ただ品質担保に十分なテスト?

人 vs AI

ソフトウェアテストで大事なことは、いきなりテストを書かず
何のためのテストか、どうやってテストするかを事前に検討する

テスト観点を教えて

1. UIテスト
 - レスポンシブデザイン：各デバイスでレイアウトや表示が正常か
 - リンクやボタンの動作：各ボタンが正しく機能し、ページ遷移が正常か
2. 機能テスト
 - お問い合わせフォーム：フォームが送信でき、入力エラーの処理が正常か
 - 資料請求機能：フォームが送信でき、入力エラーの処理が正常か
3. セキュリティテスト
 - SSL証明書の確認：HTTP接続の設定が正常か

人 vs AI

ソフトウェアテストで大事なことは、いきなりテストを書かず
何のためのテストか、何をどうやってテストするかを事前に検討する

1. UIテスト

- レスポンシブデザイン：各デバイスでレイアウトや表示が正常か
- リンクやボタンの動作：各ボタンが正しく機能し、ページ遷移が正常か

テスト設計をして

- テストケース1：各ページやセクションの表示確認
 - 期待結果：画像やテキスト、ボタンが正しい位置に配置されている
- テストケース2：ボタン「お問い合わせ」「ご相談・資料請求」の動作確認
 - 期待結果：ボタンを押すと正しいフォームやページが表示される

人 vs AI

1. UIテスト

- テストケース1：各ページやセクションの表示確認
 - 期待結果：画像やテキスト、ボタンが正しい位置に配置されている
- テストケース2：ボタン「お問い合わせ」「ご相談・資料請求」の動作確認
 - 期待結果：ボタンを押すと正しいフォームやページが表示される

テストケースを書いて

テストケース1：各ページやセクションの表示確認

- 目的：Webサイトの各ページおよびセクションの表示を確認する
- 手順
 - 1. Webサイトにアクセスする
 - 2. ページのすべてのセクションを確認する
- 期待結果：画像、テキスト、ボタンの位置とデザインが正しいこと

人 vs AI

テストケースを表にして

テストケース番号	テストケース名	目的	手順	期待結果	テスト実行日	テスト実行者
TC-01	各ページやセクションの表示確認	Webサイトの各ページおよびセクションが正しく表示されることを確認する。	1. Webサイトにアクセスする。 2. ページのすべてのセクションを確認する。	画像、テキスト、ボタンが正しい位置に表示され、デザインが崩れていない。		
TC-02	「お問い合わせ」「ご相談・資料請求」ボタンの動作確認	ボタンのクリックが適切に機能しているか確認する。	1. 「お問い合わせ」ボタンをクリック。 2. フォームページが表示されるか確認。 3. 同様に「ご相談・資料請求」ボタンを確認。	両ボタンともにクリック後、正しいフォームページに遷移する。		
TC-03	レスポンシブデザインの動作確認	さまざまなデバイスでレイアウトが正しく表示されることを確認する。	1. デスクトップ、タブレット、スマートフォンのサイズに画面を切り替える。	すべてのデバイスでコンテンツが正しく表示され、レイアウトが崩れていない。		
TC-04	お問い合わせフォームの送信確認	フォームの送信機能が正しく動作するかを確認する。	1. 「お問い合わせ」フォームにアクセス。 2. 名前、メールアドレス、内容を入力。 3. フォームを送信。	フォームが正常に送信され、成功メッセージが表示される。		
TC-05	資料請求フォームの動作確認	資料請求フォームの動作が正常であることを確認する。	1. 「ご相談・資料請求」ボタンをクリックし、フォームを表示。	フォームが正しく送信され、資料請求が正常に処理されることを確認。		
TC-06	無効な入力に対するバリデーションチェック	フォーム入力のバリデーションが正しく機能しているか確認する。	1. フォームのフィールドを空欄にして送信。 2. 無効なメールアドレスを入力し送信。 3. 長すぎる文字列を入力し送信。	エラーメッセージが正しく表示され、バリデーションが動作する。		
TC-07	ページの読み込み速度測定	ページの読み込み速度が適切であるか確認する。	1. メインページにアクセス。 2.	3秒以内にページが完全に読み込まれる。		
TC-08	同時アクセス時の負荷テスト	同時に多数のユーザーがアクセスした際のパフォーマンスを確認する。	1. 負荷テストツールを使用し、シミュレーションを実行。	多数のアクセスがあっても、サイトの動作が遅くならず、機能に支障が出ない。		
TC-09	HTTPS通信の確認	サイトがSSLで保護され、セキュアな通信が行われていることを確認する。	1. ブラウザのアドレスバーを確認し、HTTPS接続を確認。	サイトがHTTPSで安全に通信されている。		
TC-10	XSS攻撃対策の確認	XSS攻撃に対する防御が実装されているか確認する。	1. フォームにスクリプトタグを入力し、送信。 2. ページがスクリプトを実行しないことを確認。	入力されたスクリプトが実行されず、安全に処理される。		
TC-11	キーボードのみでのナビゲーション	キーボード操作のみでサイトをナビゲートできることを確認する。	1. キーボードのみでページを移動し、フォームを入力し送信する。	キーボードだけで全ページにアクセスし、		

人 vs AI

AI利用での注意点

- AIの考えた**テストが十分か**判断が必要
 - **足りないテスト**はもちろん困る
 - やる価値の低いテストを実行しても**時間の無駄**
 - **ローコード**や**ノーコード**のサービスも同様
- **ハルシネーションのリスク**

- **アイデアの土台**としてリストアップしてもらおう
- 人が考えた内容に対してFBをもらい、**気づきの機会**をAIにもらおう
 - **AIと壁打ち**

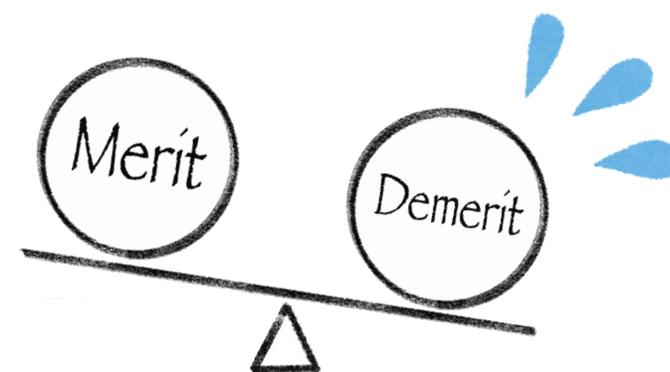
まとめ

まとめ

自動・手動の最適バランスは **ありません**

プロジェクトだから状況が一定ではない

- 開発システム / アプリケーション
- 組織 / チーム
- 新規開発 / エンハンス開発
- コスト / スケジュール
- ステークホルダー / 業界
- 開発手法



まとめ

最適バランスのためのポイント①

- 手動・自動・テストレベル（テストサイズ）を**事前に**検討する
 - テスト自動化も**開発の一つ**。**しっかりとした設計**が必要
 - 無理に**すべてを自動化**する必要はない（し、出来ない）
 - トータルで考えると**手動テストの方が早い**場合もある
- **何をテストする**のか、**どうテストする**のか、**どういう期待値**かが明確になっていないと自動テストは作れない
- 自動テストは手段の一つ。手動テストとテストとしては同じ
 - **テストケースのメンテナンス**、エビデンスはしっかり対応

まとめ

最適バランスのためのポイント②

- **再利用**できるテストを自動化する
 - **壊れる可能性は低い**が、ユーザーへの**影響が大きい**箇所
 - **サイトタイトル**や**リンク切れ**
 - **メンテが少なく**、テスト自動化の練習にも向いている
- 開発者として**気軽に変更**ができ、**UX効果の大きい**箇所はNG
 - デザイン変更
 - 自動テストのメンテナンスが大変
- 大きく作りすぎない。**少しの自動化**でも意外と楽になれる
 - 壊れたら**自動で検知できる仕組み**で、安心した開発ができる

まとめ

最適バランスのためのポイント③

- AIも十分活用できる
- しかし、AIの回答すべてを**鵜呑みにはできない**
 - 回答内容は**正しいか**
 - 品質を担保するためのテストとして**抜け漏れ**がないか
 - **ブレスト**、**壁打ち**などAIの回答を鵜呑みにしない使い方
- テストの内容の判断には**専門家の知識や経験**が必要
 - テスト慣れしている開発チームメンバー
 - QAチーム
 - 第三者検証会社

ハサミとカッター

	ハサミ	カッター
直線	素早く正確に切るのが難しい	素早く正確に切れる
曲線	正確に切りやすい	正確に切るのが難しい
下敷き	不要	状況により必要
刃の寿命と切れ味	長い。ただし、切れ味が落ちても刃の交換はできない	短い。ただし、切れ味が落ちたら刃の交換ができる
厚み	厚みがあるものは苦手	厚みがあっても切ることができる
ダンボールの封	切りにくい	切りやすい

ハサミとカッター

切りやすさ、使いやすさのほかに安全性の観点もあります。
カッターの方が使いやすい状況でも、小さな子供が使う状況では
ハサミを使わせると思います



ハサミとカッター

切りやすさ、使いやすさのほかに安全性の観点もあります。
カッターの方が使いやすい状況でも、小さな子供が使う状況ではハサミを使わせると思います

目的や状況に合わせた
住み分けが必要



GENZ