

その常識、変えてみせる。

SHIFT

人財育成は社会を救う ～最速でアジャイルなQA人財を育てる仕組み～

株式会社SHIFT

福田 麻樹

QA業界における人財とは？

QAというお仕事は社会的に見ても根幹を支えているといっても過言ではありません。
なぜなら、それは品質とは人々の財産やときには命すら左右するためです。
そして、なかなか失敗は許されない職業です。

ですがIT需要が上がるなかその**根幹**を担う**QA専門家の人財が不足**しています。

要するに

QA専門家の人財不足 = 社会のピンチである！

こちらのセッション内容は**人財育成の観点から社会のピンチを救う**ために試行錯誤したお話です。

- 1.スピーカー紹介
- 2.人材育成における課題について
- 3.意識した内容と課題へのアプローチ
- 4.育成問題を解決するため、最速の教育スキーム爆誕
- 5.今後の展望
- 6.終わりに
- 7.Q&A

01 スピーカー紹介

- 名前：福田 麻樹 (Maki Fukuda)
- 所属：株式会社SHIFT
- 役割：QAリード 兼 スクラムマスター
- 信念：楽しくなければアジャイルじゃない



02 人材育成における課題について

質問させてください。

本当に足りていないのは何か？ 足りていないのは人手ですか？



売り手市場だからといって企業が欲しているのは人手ではなく人材です。

特にQAのような責任ある業種でもっとも必要なのは人手ではなく人材です。
ですから人材は明確に社会にとっても会社にとっても**財産**となると考えています。

人手＝「働く人」

人材＝「スキルや技術を有する人」

人手 ≠ 人材
人材 = 人財 と定義しています。

代表的に以下が考えられます。

- **スカウティングや転職受け入れ**
(どこかの球団のように他社からスキルをもった人材を受け入れる)
- **自社内で育成を行い育てる**
- **いっそのことQAを諦め、開発する**



スカウティングすれば手っ取り早いのですが、今回は持続可能性を鑑み育成についての観点で仕組みを構築いたしました。

- スカウティングや転職受け入れ
(どこかの球団のように他社からスキルをもった人材を受け入れる)

ここ！

■ 自社内で育成を行い育てる

- いっそのことQAを諦め、開発する

■ 人材育成に十分な時間をかけられない「時間」

-入社して研修中でも給与が発生します、長く育成に時間をかけること、デメリットが多いため手厚く時間を取ることは難しい

■ 育成する側の指導力不足「能力不足」

-現場での仕事と育成では毛色が違う業務で専門性を要します、育成する側の指導能力を身につけることは難しい

■ 育成における仕組みや育成自体を評価する基準が無い「仕組み／基準」

-育成される側の評価（育成は十分だったか？）、指導する側も適正であったか？を評価する仕組みがなく持続可能性がない

■ QAの業界では「失敗できない」

-人は失敗から学ぶと良くいいますが品質は時に人生や人命まで左右されることもあるため、失敗が許されるケースは少なく失敗から学ぶときには時すでに遅しというパターンも。。。

03 意識した内容と課題へのアプローチ

■学習定着率の向上

ラーニングピラミッドを意識しそのなかから、**アクティブラーニング**を重視することで定着率の向上



[学ぶ] = [自ら手を動かす]+[自ら説明／アウトプット]

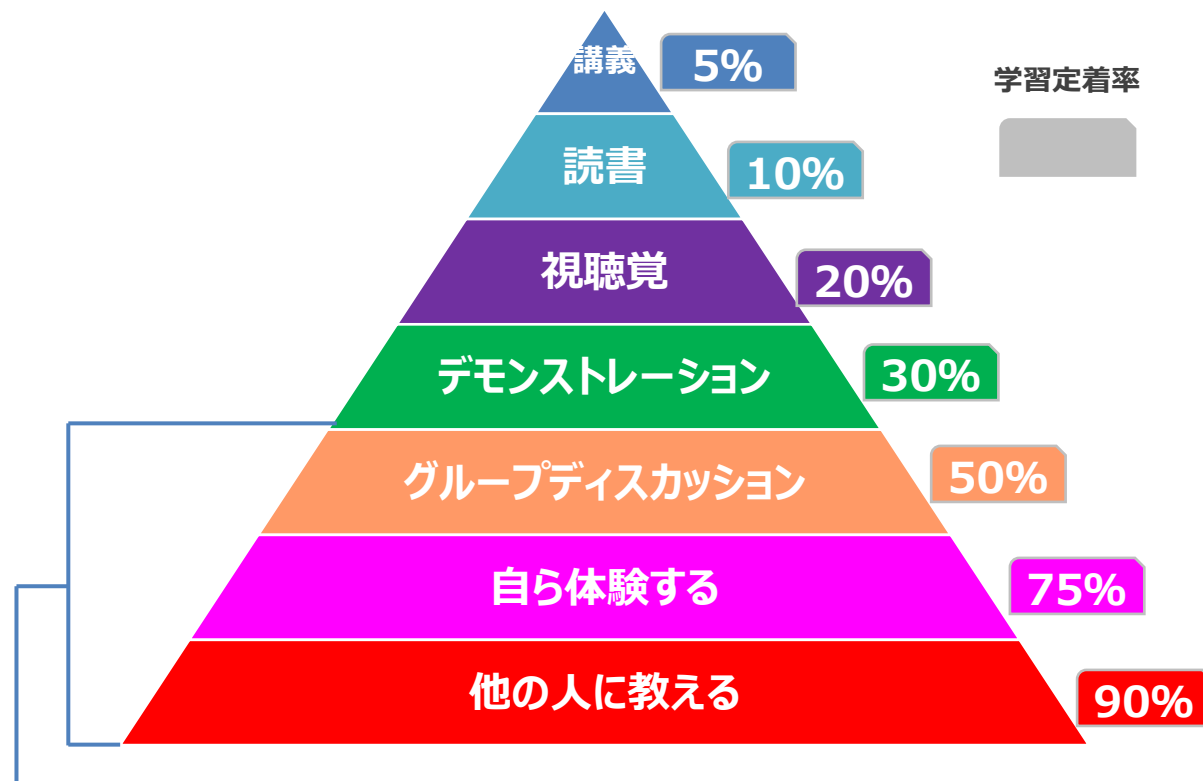
■ 学習定着率の向上

～学習しても定着しないのでは本末転倒、学習定着率を高めるスキーム考案～

アメリカ国立訓練研究所の研究によると、学習方法と平均学習定着率の関係は「**ラーニングピラミッド**」というピラミッドのような図で表すことができます。

そしてラーニングピラミッドのなかでもより能動的な「**他者と議論する**」「**実践による経験・練習**」「**他人に教える**」の3点を重視した勉強方法である**アクティブラーニング**を重視することで定着率の向上を図りました。

ラーニング・ピラミッド



アクティブ・ラーニング

■ 学習定着率の向上

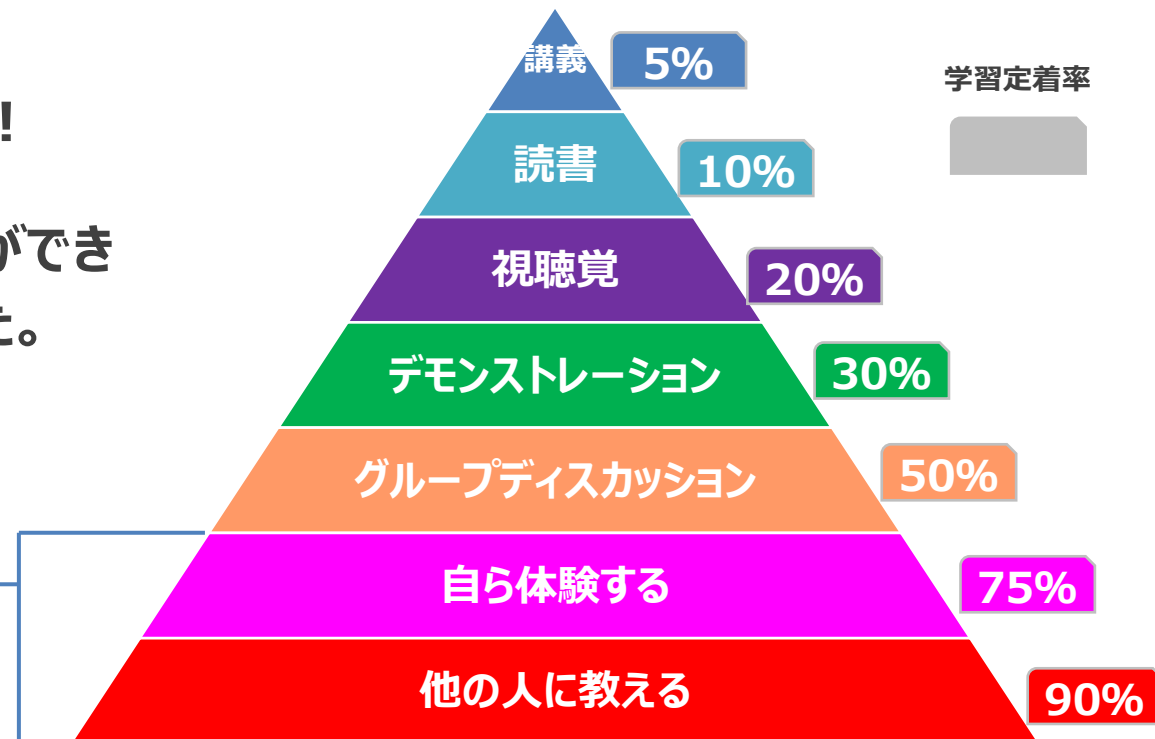
[学ぶ] = [自ら手を動かす]+[自ら説明／アウトプット]

この形式を意識することで、飛躍的に学習定着率が向上！

平均値だと90%の定着率を見込め最速の教育スキームができるのではないかと、**仮説**を立て、**検証**を行うことにしました。

ここを意識！

ラーニング・ピラミッド



■「時間」

1日1時間かつ1週間（アジャイル開発を意識）（合計で5時間！！）

■「能力不足」

育成スキームと**同時並行で受講生を講師側に育成**

■「仕組み／基準」

SHIFTではすでに**人が人によい影響を与える活動**に関して、評価へと還元される

■「失敗できない」

安全になぜか転ぶ仕組みで解決策を自ら導き出す

■ 教育に要する「時間」

実際に教えるとなると講師側と受講側の双方の時間(工数)が必要になります。
特に講師側は普段通常業務もあるため、時間を大きく割くことは難しい現状が考えられます。
また受講側もアサインされないまま勉強をつづけることもできずに中途半端な学習状況でアサインもありうる。

「時間」の問題を上記と踏まえ、**1日1時間かつ1週間**(アジャイル開発を意識)と定め実証することとしました。(合計で**5時間**ほど、少しハードルを上げすぎたかな?)
イメージするならビリーズブートキャンプのような形です。(知らないかたもいるかも)

また、学習する際時間が限られている以上、1対多数だと学習スピードが遅い人は取り残されてしまうため個々に合わせられるよう**1on1**での学習スタイルが最適としました。



※ブートキャンプとはアメリカ合衆国で「新兵訓練施設」を意味する口語表現です

■ 育成する側の指導力不足「能力不足」

この教育スキームを完成させるには講師の育成も必要不可欠です。私一人ですべてを見ることは難しいでしょう。

講師の育成について、少し話が脱線します。

山本五十六をご存じでしょうか？その名言を一部抜粋致します。

「やってみせ、言って聞かせて、させてみて、褒めてやらねば人は動かじ・・・」

この名言を思い出し、育成スキームと**同時並行で受講生を講師側に育成**すれば効率的で効果があるのではと仮説を立て、**受講者の真のゴールを講師になること**として設定し、自身の姿を見せやり方を学んでもらい、称賛する形で検証することにしました。

■ 育成における仕組みや育成自体の評価基準「仕組み／基準」

育成スキーム終了後、アンケートで受講者側の率直な意見などのフィードバックをもらい、講師側における評価を受けつねに改善できる仕組みを構築しました。

また**SHIFT**ではすでに**人が人によい影響を与える活動**に関して、上長は褒め評価へと還元される仕組みが行われており講師になる**モチベーション**の心配する必要はありませんでした。

なので、自ら上長にアピールするきっかけを与え持続可能性を生み出せます。



■ QAの業界では「失敗できない」

失敗から学ぶことは効果が高いとされていますが失敗することを見込めない、なぜなら実際では失敗できません。また、失敗したことでやる気を失ってしまうこともあれば、失敗の理由がわからないこともあります。

失敗から学習することのもっとも大きな障害は、**失敗が苦痛を伴う**ため、失敗から学ぶといっても**本当に失敗から学ぶことは実は難しい**のです。

質問させてください。

**失敗できないことに対して行う対策は何でしょうか？
【転ばぬ先の杖】なのではないでしょうか？**

■ QAの業界では失敗できない「失敗できない」

転ばぬ先の杖（知識）も大事ですが、それが必ずしも直面する失敗に効力のある処方箋とはならない。

実際私も若いころ、**教えてもらったり思っていた失敗と違う！**

といった経験があり大いに失敗しました。そのときの心のダメージはかなり大きかったです。

ではどうするか？

失敗できる環境で、**それとなく、わざとらしくなく、安全に**大いに転倒できる環境があればと考えました。

要するに**安全になぜか転ぶ仕組みで解決策を自ら導き出す**ということです。

04 育成問題を解決するため、最速の教育スキーム爆誕

かなり欲張りではありますがすべてのアプローチを実現させるため、頭を悩ませました。
そこでできた内容が次のページです。



■ 用意するもの

- 講師 : 1名~2名 (的確に鋭い抜け目のないお客様役と講師の帽子を被りわけてもらいます)
- 受講生 : 1名
- テスト対象 : 自社内の社員のみ閲覧可能なWebサイト
- 環境 : 自由にお絵描きできる環境 (Miroを使用しました)
- テスト観点一覧 : SHIFTのナレッジが蓄積された観点一覧を使用

- フォーマット類
 - 超簡易的なテスト設計書のフォーマット
 - ハイレベルテスト設計書のフォーマット (概要レベル)
 - ローレベルテスト設計フォーマット



■ 大きな流れ

- テスト対象のWebサイト
(セキュリティ観点やパターン数過多が折り込まれている)を用意
- マインドマップを書いてもらいテスト対象の体系的／潜在的な正体を暴く
- 簡易的なテスト計画を立て、仮想のお客様へのレビュー体験
(フィードバック厳しめ)
- 改善ポイントを伝え再チャレンジ
- テスト設計を完了させるまでを疑似体験

■ カリキュラム内容（概要）

➤ LESSON① マインドマップでテスト対象の正体を暴こう

～テスト対象から何のために使うもので、何ができていないといけないのかを書き出します。ここでどんな観点のテストをする必要があるのかまで考えてもらいます～

狙い：テスト対象の正体を知ることの大事さと下地を育みます、実際マインドマップでなくてもよいが今回はマインドマップで洗い出します

➤ LESSON② テスト計画(簡易)を書いてみよう

～なぜ？どこで？どの環境で？やらない環境は？なんでやらないの？などを書き出しエビデンスを作成してもらいます～

狙い：テスト設計での合意形成のエビデンス化の下地を育みます、やれない箇所や環境が記載されているか？なぜやらないか？に対する説明力がつきます

➤ LESSON③ ハイレベルテスト設計（概要設計）を書いてみよう

～SHIFTの標準観点をういて観点を洗い出し、観点と簡易的な手順並びに期待結果を書き出してもらいます～

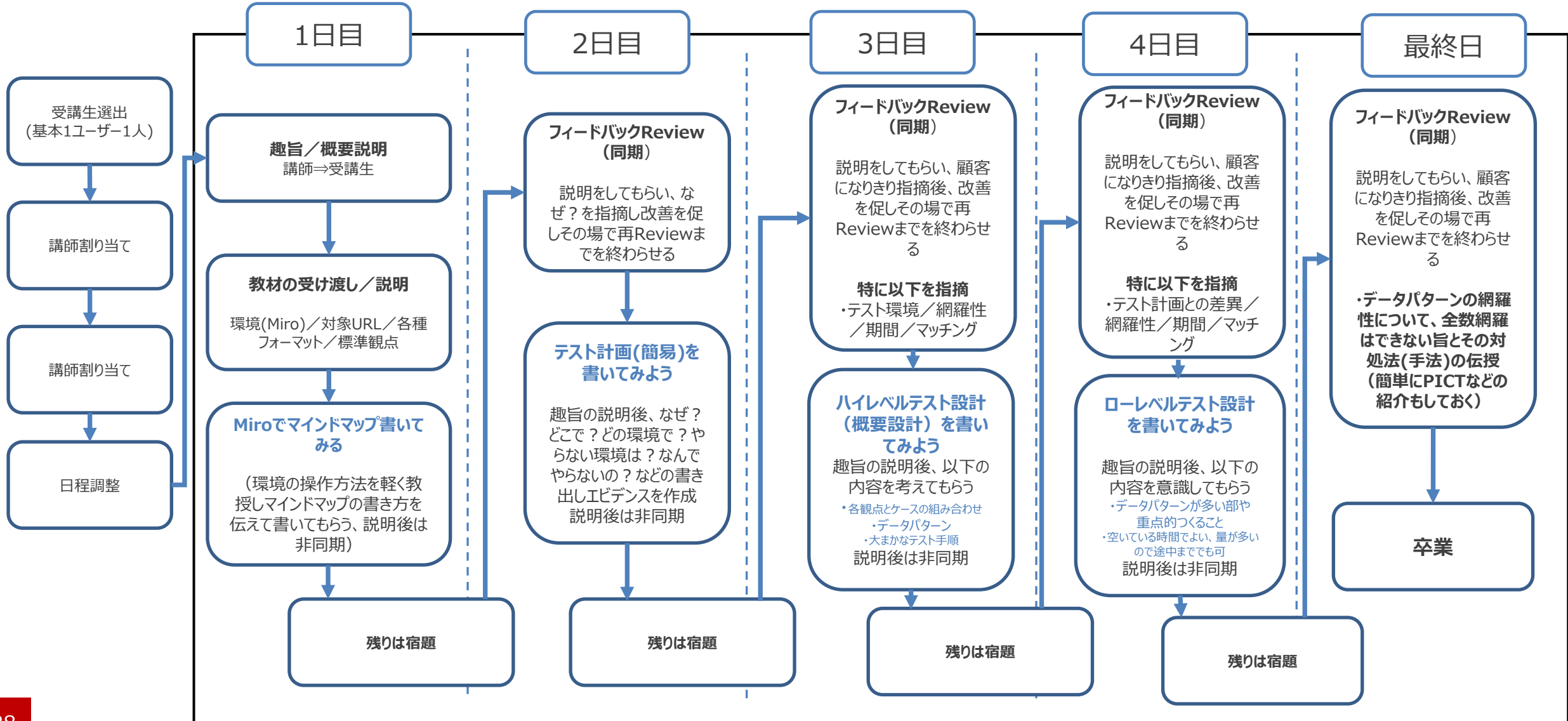
狙い：実践形式でテスト設計力を育みます、SHIFT標準観点をういて使用感になれる狙いとテスト対象と観点の結節を短時間で行うトレーニングです

➤ LESSON④ ローレベルテスト設計を書いてみよう

～実際のテストフォーマットに落とし込みパターン表などを構築します～

狙い：最終的にハイレベルテストケースからローレベルに落とし込む作業を行い、案件へのアサインが行えるようにします

■ スキーム

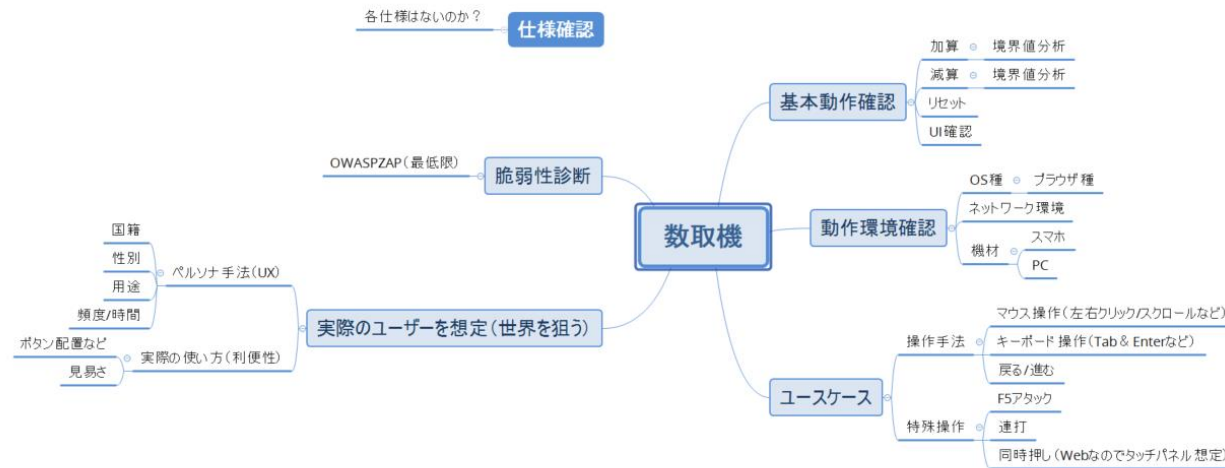


■ LESSON① マインドマップでテスト対象の正体を暴こう

テスト対象から何のために使うもので、何ができていないといけないのかを書き出します。ここでどんな観点のテストをする必要があるのかまで考えてもらいます

狙い：テスト対象の正体を知ることの大事さと下地を育みます、実際マインドマップでなくてもよいが今回はマインドマップで洗い出します。

以下の例図のようにMiroを用いてマインドマップを記載します。



■ LESSON② テスト計画(簡易)を書いてみよう

なぜ？どこで？どの環境で？やらない環境は？なんでやらないの？などを書き出しエビデンスを作成してもらいます

狙い：テスト設計での合意形成のエビデンス化の下地を育みます、やれない箇所や環境が記載されているか？なぜやらないか？に対するの説明力がつきます

以下の例図のようにExcelファイルを用いて記載してもらいます

- ・テスト目的（目的／概要／PBI s (ないけど)）
- ・テスト方針（動作環境／対象アプリケーション／確認方法／テスト対象）
- ・テスト内容（U I ／機能／非機能／テスト環境／対象端末／対象外とする端末／テストの制約／リスクと対策）

テスト目的	
・テスト目的	
・テスト概要	
・対象となるPBI s	
テスト方針	
動作環境 対象アプリケーション 確認方法	
テスト対象	

テスト内容	
テストで見えるべきポイント	
[重要テスト]	
[編集テスト]	
テスト環境	
対象の端末	OS ブラウザ
対象外の端末	
テストの制約条件、前提条件	
リスクと対策	

■ LESSON③ ハイレベルテスト設計（概要設計）を書いてみよう

SHIFTの標準観点をを用いて観点を洗い出し、観点と簡易的な手順並びに期待結果を書き出してもらいます。

狙い： 実践形式でテスト設計力を育みます、SHIFT標準観点をを用いて使用感になれる狙いとテスト対象と観点の結節を短時間で行うトレーニングです

以下の例図のようにExcelファイルを用いて記載してもらいます

- 基本的にはSHIFT標準観点表を用いて書き出してもらう。
- トレースが追えるようにIDも控える（福田推奨）
- 簡単な操作概要、期待結果を記載する
- ここに書いたものを基本的にローレベルにそのままコピーし広げると完成できるレベルで書いてもらう

※できるところまで完璧は求めない（時間の都合上）

テスト対象画面	確認観点			操作概要	期待結果	トレース用観 点No	備考
	大項目	中項目	小項目				
ヒトログ_メンバー_選択画面	UI	写真表示	-	画面表示し確認	写真表示に問題がない事（自分の写真で確認） 写真ない場合は男性アイコン/女性アイコンが性別通りに表示されている事	※※※	
ヒトログ_メンバー_選択画面	機能テスト	マウスオーバー	-	各メンバー写真（アイコン）にマウスオーバーを行い確認	マウスオーバー時にメンバーの概要が表示される メンバーの概要表示に不備がない事を確認	※※※	
ヒトログ_メンバー画面	非機能テスト	インフォメーションボタン	連打	外面内に表示されているインフォメーションボタンを連打	連打を行っても動作に問題がない事を確認	※※※	

■ LESSON④ ローレベルテスト設計を書いてみよう

実際のテストフォーマットに落とし込みパターン表などを構築します

狙い：最終的にハイレベルテストケースからローレベルに落とし込む作業を行い、案件へのアサインが行えるようにします。

実際のテストシートフォーマットを利用しリアルに設計してもらいます

・更新履歴の記載

ドキュメント名	テスト設計書	作成日	2020/3/1	最終更新日	2020/8/27
テスト設計書名	結合テスト設計書_IT001_○○○○○機能	作成者	○○○	更新者	△△△
プロジェクト名	○○○○○検証	テストレベル	○○○テスト	テストタイプ	○○○○○テスト
プロダクト名	XXXXXXXXXXXXX	サブ名			
更新日	バージョン	更新者	更新内容		
2018/3/1	1.0.0	○○○	初期バージョン作成		

・テスト設計

確認項目 No	テスト対象 (テスト区分1)	テスト対象 (テスト区分2)	テスト対象 (テスト区分3)	テスト対象 (テスト区分4)	テスト実行手順	PTN	テスト観点	確認項目	期待値	備考	要件ID (任意)	実施要否	Excel 集計用 実行結果
1													
2													
3													
4													

・実行パターン表入力

■ 所属コード存在エラーチェックパターン

#	因子	水準	備考	分類 (任意)
P1	所属コード	存在するコード	コード値:「1234」を入力	
P1	所属コード	存在しないコード	所属マス外に存在しないコード コード値:「9999」を入力	
P1	所属コード	未設定	-	
P1	確認項目	期待値	-	
P1	所属コードがマス外に存在しない場合、エラーとなること		「E001」エラーメッセージが表示されないこと	
P1	所属コードがマス外に存在しない場合、エラーとなること		「E001」エラーメッセージが表示されること	
END	END	END	END	END

非常に暖かい声、率直な改善点などのお声をいただいております！！
blogの記事などにまとめてもらったりと受講生からは好評でした！ご興味ある方は一読ください！



新入社員は『テスト設計ブーツキャンプ』で何を学んだのか

♡ 58

SHIFT Group 技術ブログ
2023年3月1日 19:36



Q4 卒業後、実際にプロジェクトにアサインして、ブーツキャンプで学んだことは活かされていますか？

A

ブーツキャンプのおかげで実際に案件でテスト設計をする際に、ブーツキャンプでの経験がとても活きていると感じます。現在、業務の8割がテスト設計の案件にアサインされています。

ブーツキャンプではテスト計画、テスト設計のスキルと考え方を学ぶことができました。このスキルはテストエンジニアとして実務で必要とされる場面が多々あります。

アサイン後、自信がついたことでほとんど不安を感じなかったため、学べて良かったと改めて感じました。また、ブーツキャンプの内容が本当にすぐ使うことができる実務向けの内容だったことも実際にアサインされてみて感じました。

blog記事：<https://note.shiftinc.jp/n/n67de46858f06>



受講後の声を紹介します。

➤ マンツーマンで親近感がある

・マンツーマンで親近感があるため質問しやすい・わかりやすい・細かく丁寧に教えてもらえる

➤ 実践的なカリキュラムで、レッスンの順番がテスト設計を行う上での思考フローになっている

実践的なカリキュラムで、レッスンの順番がテスト設計を行う上での思考フローになっている点が優れていると感じました。このブートキャンプの内容を手癖に出来ると自然にテスト設計が出来るようになるかなと思います。なので、一度ブートキャンプでやり方を体験した後、案件配属後などにブートキャンプを思い出しながら実際にテスト設計することで、血肉になっていくものだと感じました。来週から実際にテスト設計をしなくてはならない人を短期間で育て上げることにフォーカスした内容で非常に有意義でした。

➤ 講師が実際に現場での経験を基にエッジの効いた質問をしてくれます。

テスト設計の勉強会に出たり、実務でテスト設計はしていたので正直初めは「いや余裕やろw」と思っていたのですが、自分が思いつかなかった視点から次々に質問され鼻っ柱を折られました…。講師が実際に現場での経験を基にエッジの効いた質問をしてくれます。現場で失敗する前にここで失敗できるので、非常に貴重なチャンスだと思います。毎回新しい気づきを得られ、とても有意義でした！ありがとうございました！！

ここでは紹介しきれないほど、お声をもらってます

テスト設計においては少しはできるだろうと思っていましたが、実際に行ってみるとあーでもないこーでもないと迷っている時間が多く、想像していたよりも失敗が多かったです。ブートキャンプを通して失敗した点については今後改善活動を行ういいきっかけになったと思います。joinしている案件ではまだテスト設計は行っていませんが、実際にテスト設計を行う際には役立てたいと思います。

新しいQAの考え方の基礎を教えてくださいましたと思います。旧来の機能仕様書から全機能フルピックアップ/全件テストの発想からは隔絶の感があります。テスト担当者がどう受け取るかを丁寧に考えないといけないのも、テスト担当と開発担当が明確に分業しているからなのだとわかりました。自分がもう少し時間を使ってじっくりと取り組めたら良かったのです。

05 今後の展望

- さらなるレベルアップを求める声に対応（受講生からのリクエスト）
もっといろいろなケース（難しいケース）をこの方式で学びたい！

■ 展望

現在のスキームをFoundationとして、さらにAdvance版のリリースに向け稼働中

盛り込みたいもの

- こんなときどうする？ 状況別のテスト設計方法（リスクベースドなど）
- 現場で使える、さまざまなテスト手法

■ 課題

現状では受講後の評価基準が定性評価のみの指標しかない、定量的に計りたい

■ 今後の対応策は以下を模索中

- ・受講後アサイン先案件内での上長評価（点数式）
- ・受講生の単価を指標とし、伸び率を計測
- ・離職率の計測

06 終わりに

みなさま、いかがでしたでしょうか？

このカリキュラムでは必要以上に知識や内容について、教えない形で構築しました。

カリキュラム後も自ずと吸収し活躍できる人財へと成長してほしいと願いを込め、あえてティーチングではなく**コーチング**の技術を取り入れ自ら気づき自己成長を促す形としています。

実際このカリキュラムを実践してもらい、IT未経験な方がアジャイルの短いスパンでの開発下でテスト設計者として活躍できるようになりました。実際にコーチとして動いてくださる方もできてきております。

セッションの感想やご質問などをお待ちしております！

JaSST'24 Kansai セッションアンケート



ご清聴ありがとうございました

その常識、変えてみせる。

SHIFT