



テスト自動化でも変わらない、テストの大切なこと

koyaman / JaSST'20 Tokai

この発表のゴール

テスト自動化でも変わらない
大切なことを
皆さんが認識すること。



本日の流れ

- おっさん誰？
- テストの目的と本質の話
- テスト自動化スキルの話
- まとめ



Section
00

おっさん誰？



QAアニキ:コヤマン

小山 竜治

Ryuji Koyama

@koyaman2

<https://github.com/koyaman2>

PublicAPI/新認証基盤などのテストを担当

ソフトウェア品質屋 / テストマネージャー / 自動化エンジニア

元海上J官(面影ナシ..)→ブラック派遣企業(転職・転籍)
→富士ゼロックス子会社の契約社員(転職)→社員(転職)
→カスペルスキー(転職)→freee(転職)。

アジャイルチームに入ってからテストや品質戦略などなど、
「何もないところからいいカンジにテストする仕組みを作る」
ことを生業としている。

組み込み系システム(複合機、医療機器)からクラサバシステム、
B2Bセキュリティソリューションなどを経て

Web系クラウド会計のfreeeでアジャイルテストを実践。

日本科学技術連盟にてソフトウェアテスト資格試験公認トレーニングコース講師、
テスト自動化研究会やSeleniumConfTokyoなどの活動。
ASTER(ソフトウェアテスト技術振興協会)正会員。

犬好きのおっさんです。

色々なテストプロジェクト
をやってきた
何もないところに入って
テストのやり方を構築するマン

なぜ構築マンが テスト自動化の話？

AutomationTest.SSF

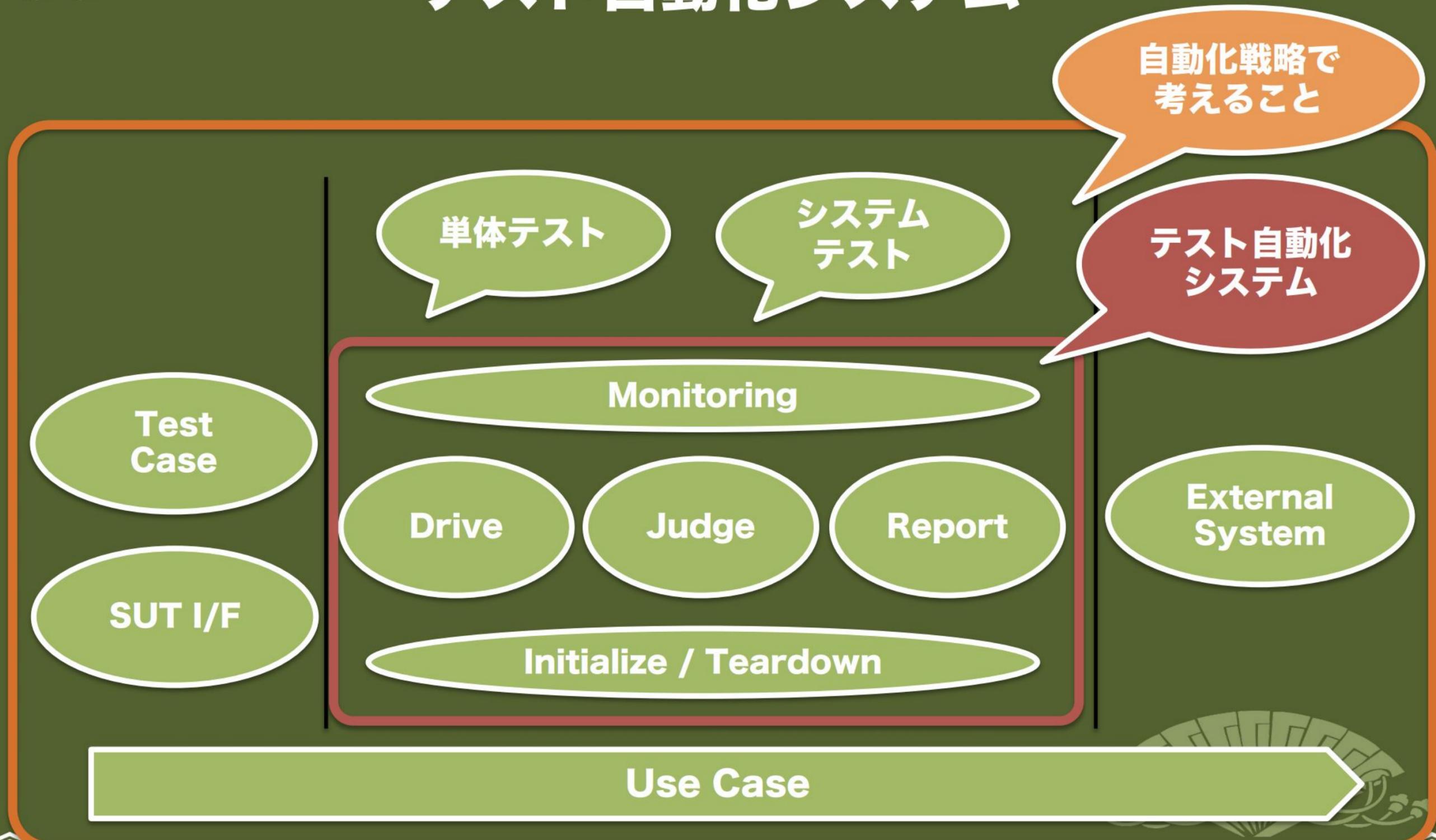
テスト自動化における
スキルの定義をしたもの
Automation Test
Skill Standards Framework

**テスト自動化研究会の
皆で作りました。**

全体像



テスト自動化システム





Section
01

テストの目的と本質の話
—そもそも
「なぜテストするんですか？」

どんな現場でも使えるテストの目的の定義



色々な現場に入ってきましたが、どこでも使える
JSTQB(旧シラバス)の「テストの目的」の定義

- ❖ 欠陥を抽出する
- ❖ 対象ソフトウェアの品質レベルが十分であることを確認する
- ❖ 意志決定のための情報を示す
- ❖ 欠陥の作りこみを防ぐ

※新シラバス(2018版)ではこれをもう少し細かく書いていますが、
抽象的にいうと上の4つで間違いないです。

(補足)ISTQBシラバス(2018)の「テスト目的」引用



1.1.1 テストに共通する目的

すべてのプロジェクトで、テストの目的は以下を含む

- ・欠陥を防ぐため、要件、ユースケース、設計、およびコードなどの作業成果物を評価する
- ・明確にしたすべての要件を満たしていることを検証する
- ・テスト対象が完成したことを確認し、ユーザーやその他ステークホルダーの期待通りの動作内容であることの妥当性確認をする
- ・テスト対象の品質に対する信頼を積み重ねて、所定のレベルにあることを確認する
- ・欠陥や故障を発見し、ソフトウェアの品質が不適切になるリスクレベルを低減する
- ・ステークホルダーが意志決定できる、特にテスト対象の品質レベルについての十分な情報を提供する。
- ・契約上、法律上、または規制上の要件または標準を遵守する、そして／またはテスト対象がそのような要件または標準に準拠していることを検証する

テストの本質



知ること。

そして知ったことを伝える(=意思決定のための情報を示す)こと。

逆に言うと「知ること」しかできません。

知ったことを伝えて利用することでさまざまな活用ができますが

- 問題や欠陥があることを欠陥レポートで伝え、それを修正・是正する
- 不明点を明確にして伝え、それを改善したりする

テスト自体はあくまで「知ること」しかできない。

さて。では何を知りたいんですか？

JSTQBのテストの7原則



- ❖ テストは欠陥があることは示せるが、欠陥がないことは示せない
 - 悪魔の証明だぜ
- ❖ 全数テストは不可能
 - 「全部テスト」は非現実的だぜ
- ❖ 初期テストで時間とコストを節約
 - 早めからやろうぜ
- ❖ 欠陥の偏在
 - 偏るんだぜ
- ❖ 殺虫剤のパラドックスにご用心
 - 同じ写真じゃ新しいところに気付けないぜ
- ❖ テストは状況次第
 - モノや状況によるんだぜ
- ❖ 「バグゼロ」の落とし穴
 - 大事なのはそこじゃあないぜ

SWEBOKのソフトウェアテストの定義



SWEBOK—SoftWare Engineering Body Of Knowledge—
ソフトウェアエンジニアリング基礎知識体系より引用

第5章 ソフトウェアテスト

ソフトウェアテストとは、通常は無限に大きいと考えられる
「プログラムの振る舞いの実行領域」から
最適だと考えられる有限な「テストケースの集合」を選定し
所期の通りかどうかを実際に動かして検証すること

おおまかな目的ごとに変わる「知りたいこと」



- ❖ 欠陥を摘出する
 - 欠陥がまだありそうか？ 出なそうか？

- ❖ 対象ソフトウェアの品質レベルが十分であることを確認する
 - 想定通り／予定通りに動くか？

- ❖ 意志決定のための情報を示す
 - 何を知ったのか？ どんなことが起こったのか？

- ❖ 欠陥の作りこみを防ぐ
 - ヤバそうなところ、ヤバそうなことは何か？

状況や対象によっても目的や知りたいことが変わる



- ❖ テストは欠陥があることは示せるが、欠陥がないことは示せない
 - 悪魔の証明だぜ
- ❖ 全数テストは不可能
 - 「全部テスト」は非現実的だぜ
- ❖ 初期テストで時間とコストを節約
 - 早めからやろうぜ
- ❖ 欠陥の偏在
 - 偏るんだぜ
- ❖ 殺虫剤のパラドックスにご用心
 - 同じ写真じゃ新しいところに気付けないぜ
- ❖ テストは状況次第
 - モノや状況によるんだぜ
- ❖ 「バグゼロ」の落とし穴
 - 大事なのはそこじゃあないぜ

皆さんの関わりはどのような状況ですか？



❖ 皆さんの現場の開発プロセスは？

- ウォーターフォールですか？
- イテレーティブもしくはインクリメンタルですか？
- よくわからんけどアジャイルですか？
- リーンですか？

皆さんの関わりはどのような状況ですか？



- ❖ 皆さんはどこで絡んでいますか？
 - 要件定義ですか？
 - 開発ですか？
 - テストですか？
 - 運用・保守ですか？

皆さんの関わりはどのような状況ですか？



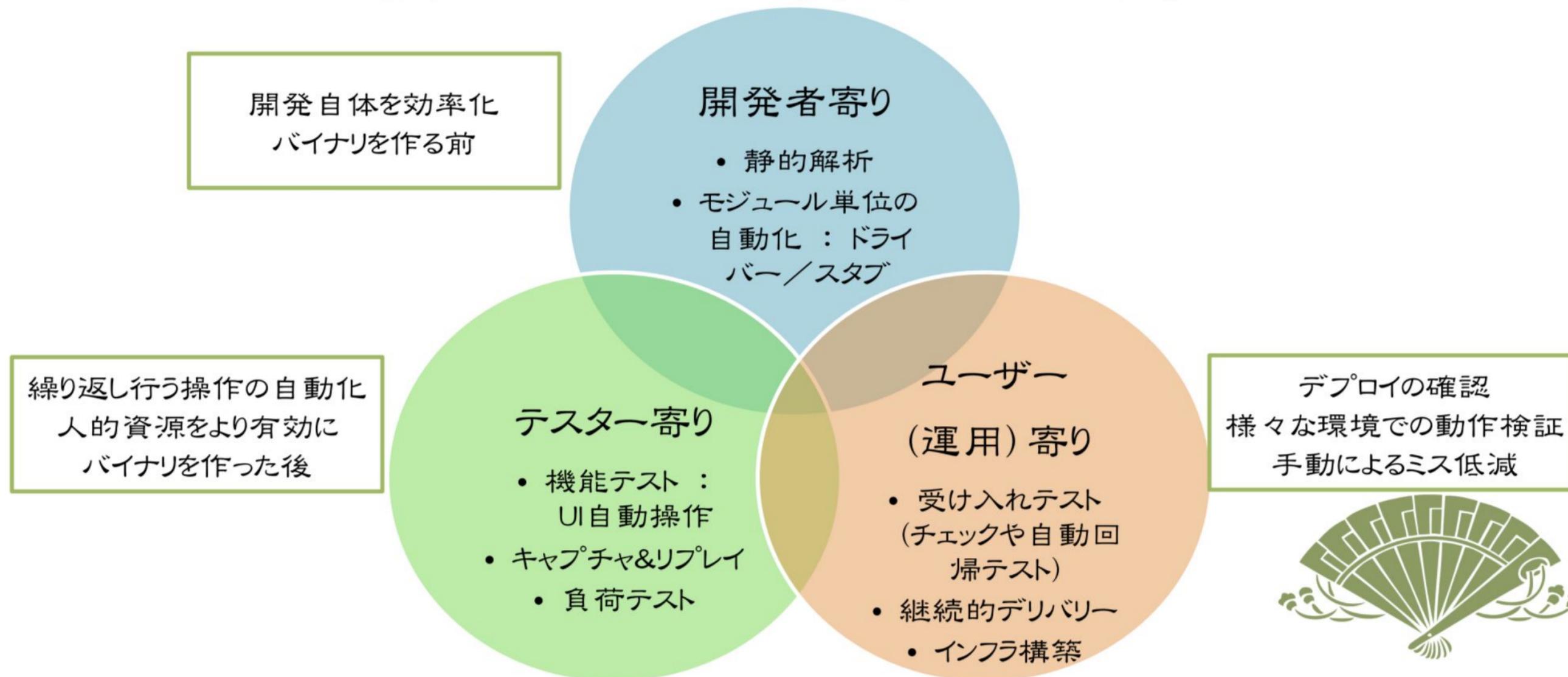
- ❖ 皆さんがテストするテスト対象のジャンルは？
 - 医療系など、人命に関わるもの？
 - 金融系など、経済的インパクトが甚大なもの？
 - 人工衛星など、社会的意義が高いもの？
 - 便利アプリ的なもの？

**自分自身の状況(コンテキスト)を
認識できましたか？**

さて。ではなぜ
テストを自動化したいんですか？

そもそもテスト自動化とは？

- テスト自動化にはいくつか種類があります。



テスト自動化の8原則(テスト自動化研究会定義)



- ❖ 手動テストはなくなるならない
 - ユーザビリティテストやぽちぽち確認はするよね
- ❖ 手動で行って効果のないテストを自動化しても無駄である
 - ゴミを自動化してもゴミ
- ❖ 自動テストは書いたことしかテストしない
 - コードなので書いたとおりに動きます
- ❖ テスト自動化の効用はコスト削減だけではない
 - セーフティネットとしての役割が多め
- ❖ 自動テストシステムの開発は継続的に行うものである
 - 自動テストシステムはサービスなので継続的に改善が必要
- ❖ 自動化検討はプロジェクト初期から
 - 最初から自動化することを考えて進めるとコスパが良い
- ❖ 自動テストで新種のバグが見つかることは稀である
 - ファジングや保守運用で見つかることはあるが、基本的には確認したいことしか確認できない
- ❖ テスト結果分析という新たなタスクが生まれる
 - いつからFailしてるのか？なぜFailしたのか？といった調査が必要なので工夫が必要

テスト自動化の利点(システムテスト自動化標準ガイド)



- ❖ **プログラムの新しいバージョンで既存の回帰テストを動かす**
 - こまめに退行(リグレッション・デグレ)確認ができる
- ❖ **テストをもっと頻繁に、たくさん行う**
 - 自動化すると頻繁に実行ができるようになる
- ❖ **手動では困難または不可能だったテストを行う**
 - 負荷テストなど
- ❖ **リソースの有効活用**
 - 夜や週末などでもテストできる
- ❖ **テストの一貫性と再現性**
 - 自動実行なので手動より正確に繰り返せる
- ❖ **テストの再利用**
 - 手動よりも繰り返しやすい
- ❖ **市場に早く提供できる**
 - 一度テストが自動化されると手動よりも早く繰り返され、結果的にリリースが早くなる
- ❖ **自信が持てる**
 - 広範囲な自動テストが成功していることを知っていると、リリース後まずい事態が起こらない自信が持てる

テスト自動化に共通する問題(システムテスト自動化標準ガイド)



- ❖ **非現実的な期待**
 - 万能を期待される
- ❖ **まずいテストिंगのやり方**
 - カオスを自動化してもカオスが早くなるだけ
- ❖ **自動テストは新しい欠陥をたくさん見つけてくれるのではないかという期待**
 - 動作保証をする目的であって欠陥を抽出する目的ではない
- ❖ **安全に対する誤った意識**
 - ずっとpassしている場合、テスト自身が間違っているかもしれない
- ❖ **自動テストのメンテナンス**
 - テスト対象が変化すれば自動テストの変化も必要になる
- ❖ **技術的な問題**
 - ツールはトレーニングが必要だし、そもそもテスト対象自体がテストしづらいとダメ
- ❖ **組織の問題**
 - インフラとして投資が必要であることを組織が理解する必要がある

テスト自動化 温故知新 (閑話休題)

システムテスト自動化 標準ガイド(2014リリース)

原著

Software Test Automation

原著

**Software Test Automation
(1999年リリース)-21年前**

テスト自動化に共通する問題(21年前の本に書いてあること)



❖ 非現実的な期待

- 万能を期待される

❖ まずいテストिंगのやり方

- カオスを自動化してもカオスが早くなるだけ

❖ 自動テストは新しい欠陥をたくさん見つけてくれるのではないかという期待

- 動作保証をする目的であって欠陥を抽出する目的ではない

❖ 安全に対する誤った意識

- ずっとpassしている場合、テスト自身が間違っているかもしれない

❖ 自動テストのメンテナンス

- テスト対象が変化すれば自動テストの変化も必要になる

❖ 技術的な問題

- ツールはトレーニングが必要だし、そもそもテスト対象自体がテストしづらいとダメ

❖ 組織の問題

- インフラとして投資が必要であることを組織が理解する必要がある



本日の流れ(再掲)

- おっさん誰？
- テストの目的と本質の話
- テスト自動化スキルの話(←イマココ)
- まとめ



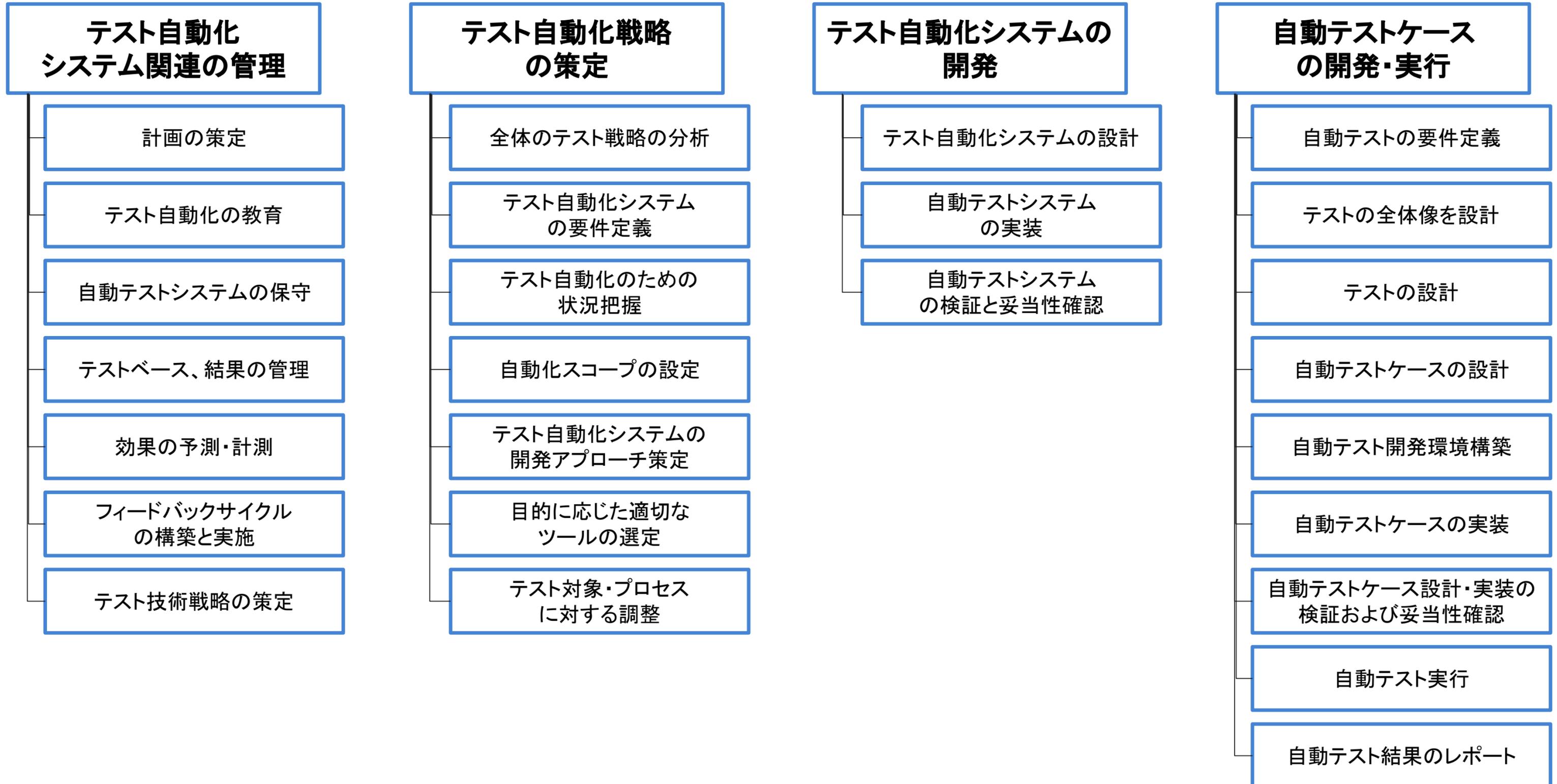
Section
02

テスト自動化スキルの話
-AutomationTest.SSFの話

全体像



AutomationTest.SSF v1.0



何が嬉しいのか？

- STARが本基準を作成する目的：

現場の人が「学習するための指標」を作れるようにする。

- 自分は何が弱いのか
- 自分は何をこれから学習 / 勉強すればよいのか
- 自分の役割にはどのようなスキルが必要なのか
をわかるようにする



**昨今、テスト自動化が
重要視されている理由**

**(ビジネスの)速度が
(さらに)必要になってきた**

適者生存



変化に対応できるものが市場でも生存する

→変化を受け入れる体制が必要

→変化を受け入れる開発プロセスである**アジャイルの台頭**

アジャイルでは常に動くものを担保し続ける

→繰り返しテストができる環境が必要

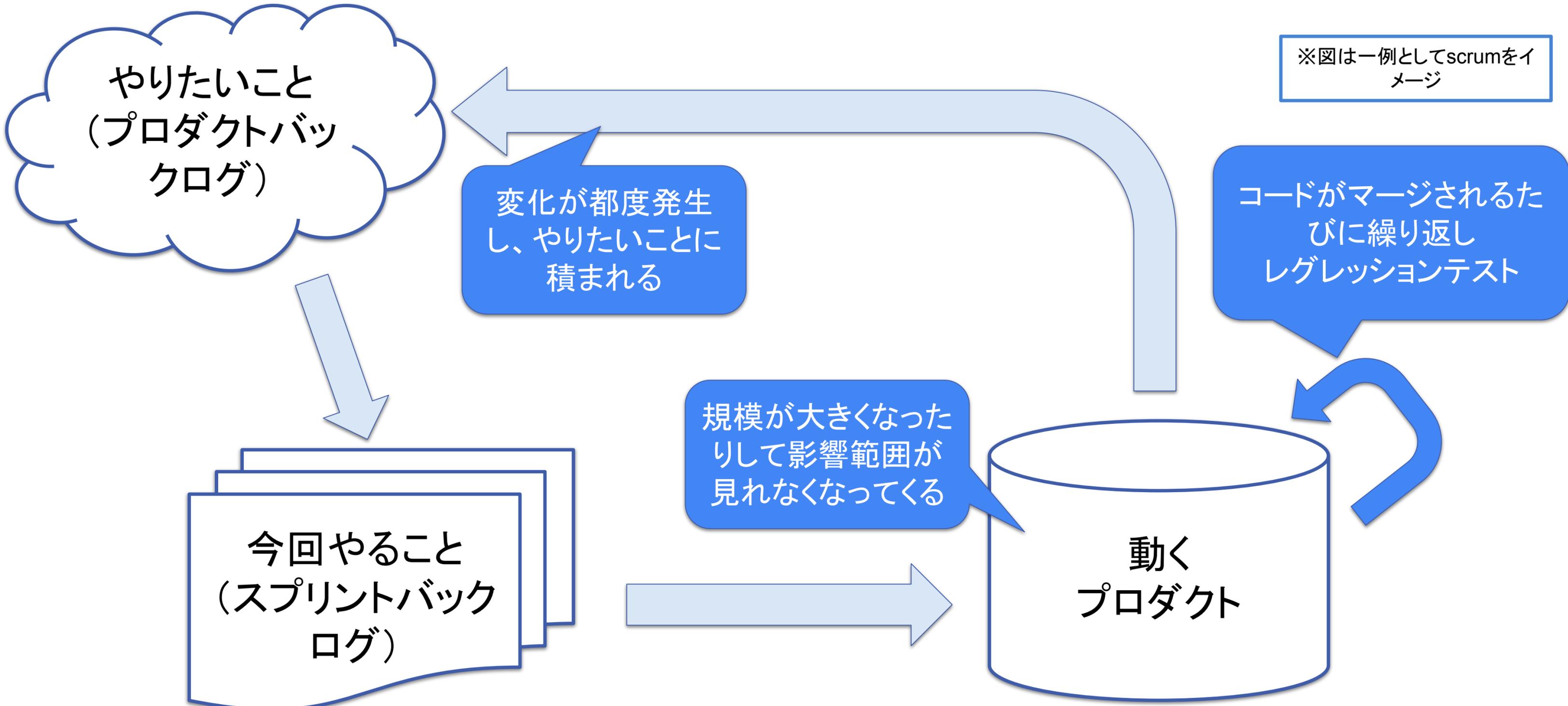
→**自動テストの台頭**

アジャイル開発には自動テストがフィットする

アジャイル開発には自動テストがフィットする



※図は一例としてscrumをイメージ



テスト自動化の利点(システムテスト自動化標準ガイド)



- ❖ **プログラムの新しいバージョンで既存の回帰テストを動かす**
 - こまめに退行(リグレッション・デグレ)確認ができる
- ❖ **テストをもっと頻繁に、たくさん行う**
 - 自動化すると頻繁に実行ができるようになる
- ❖ **手動では困難または不可能だったテストを行う**
 - 負荷テストなど
- ❖ **リソースの有効活用**
 - 夜や週末などでもテストできる
- ❖ **テストの一貫性と再現性**
 - 自動実行なので手動より正確に繰り返せる
- ❖ **テストの再利用**
 - 手動よりも繰り返しやすい
- ❖ **市場に早く提供できる**
 - 一度テストが自動化されると手動よりも早く繰り返され、結果的にリリースが早くなる
- ❖ **自信が持てる**
 - 広範囲な自動テストが成功していることを知っている、リリース後まずい事態が起こらない自信が持てる

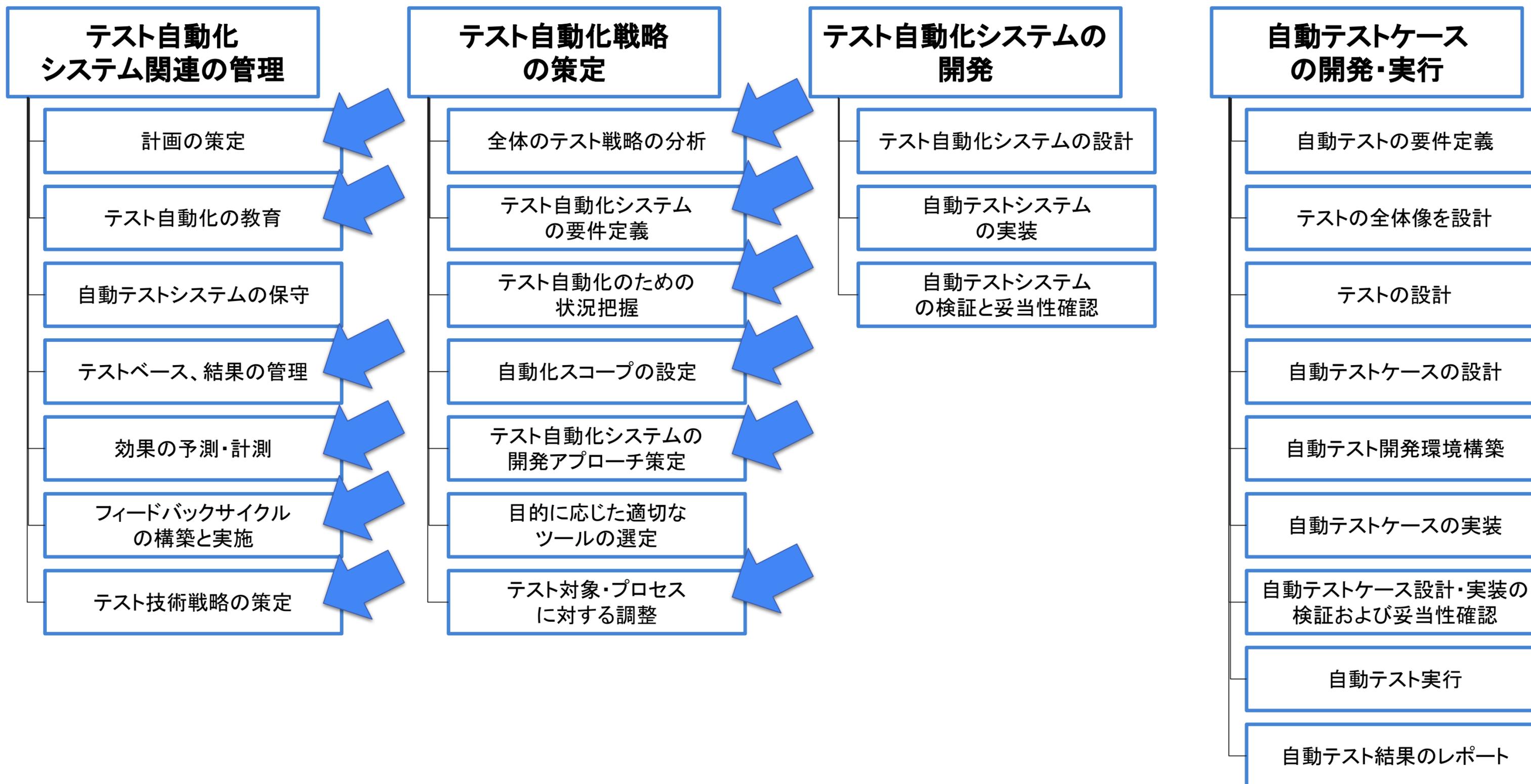
テスト自動化の大変なところ



- ❖ いつ構築するの？
 - テスト対象に合わせてみよう
- ❖ どうやって構築するの？
 - いいカンジに構築しましょう
- ❖ 何をテストするの？
 - 自分で考えましょう
- ❖ いつまでやるの？
 - ずっとやるんですよ

>いつ構築するの？
テスト対象に合わせてみよう

Automation Test.SSF v1.0でいうところの辺

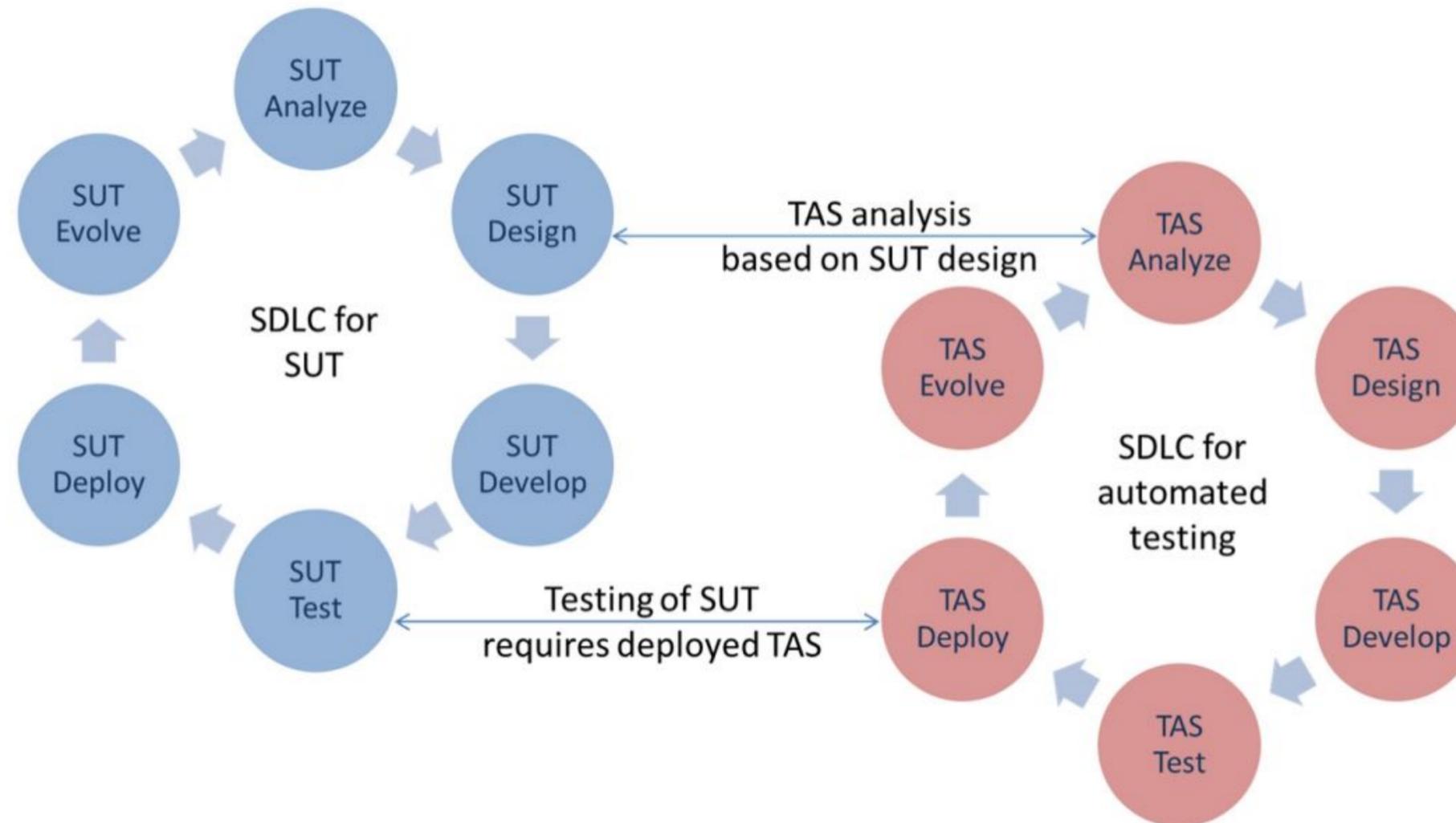


テスト対象に合わせて構築しましょう



もしまだテスト対象を開発中なのであればテスト対象の開発ライフサイクルに合わせて構築するようにしましょう

(SUT: System Under Test = テスト対象、TAS: Test Automation System = テスト自動化システムの略)



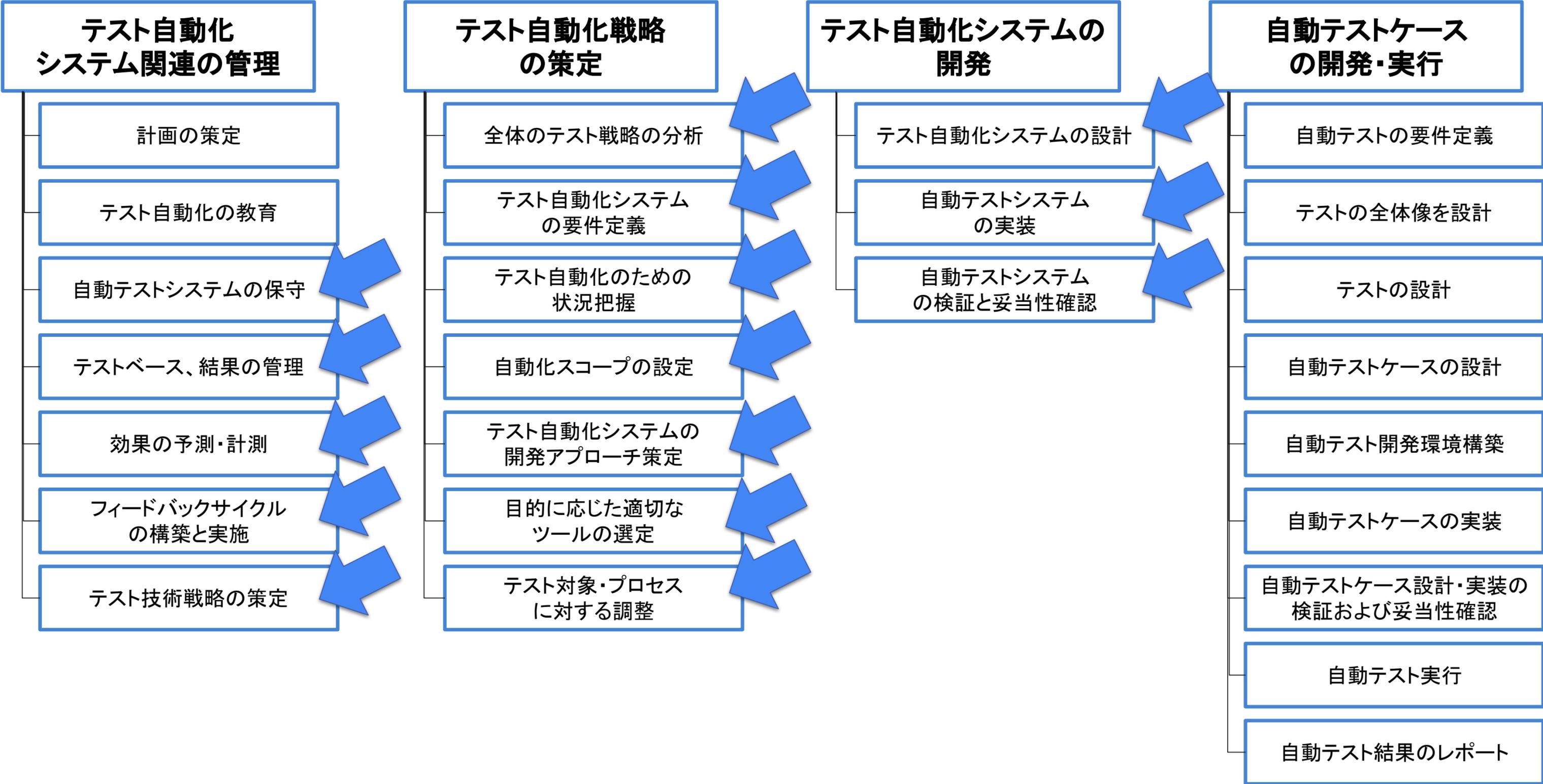
テスト対象に合わせて構築しましょう

もうテスト対象が動いている運用状況なら・・・早く作って早く安心しましょう

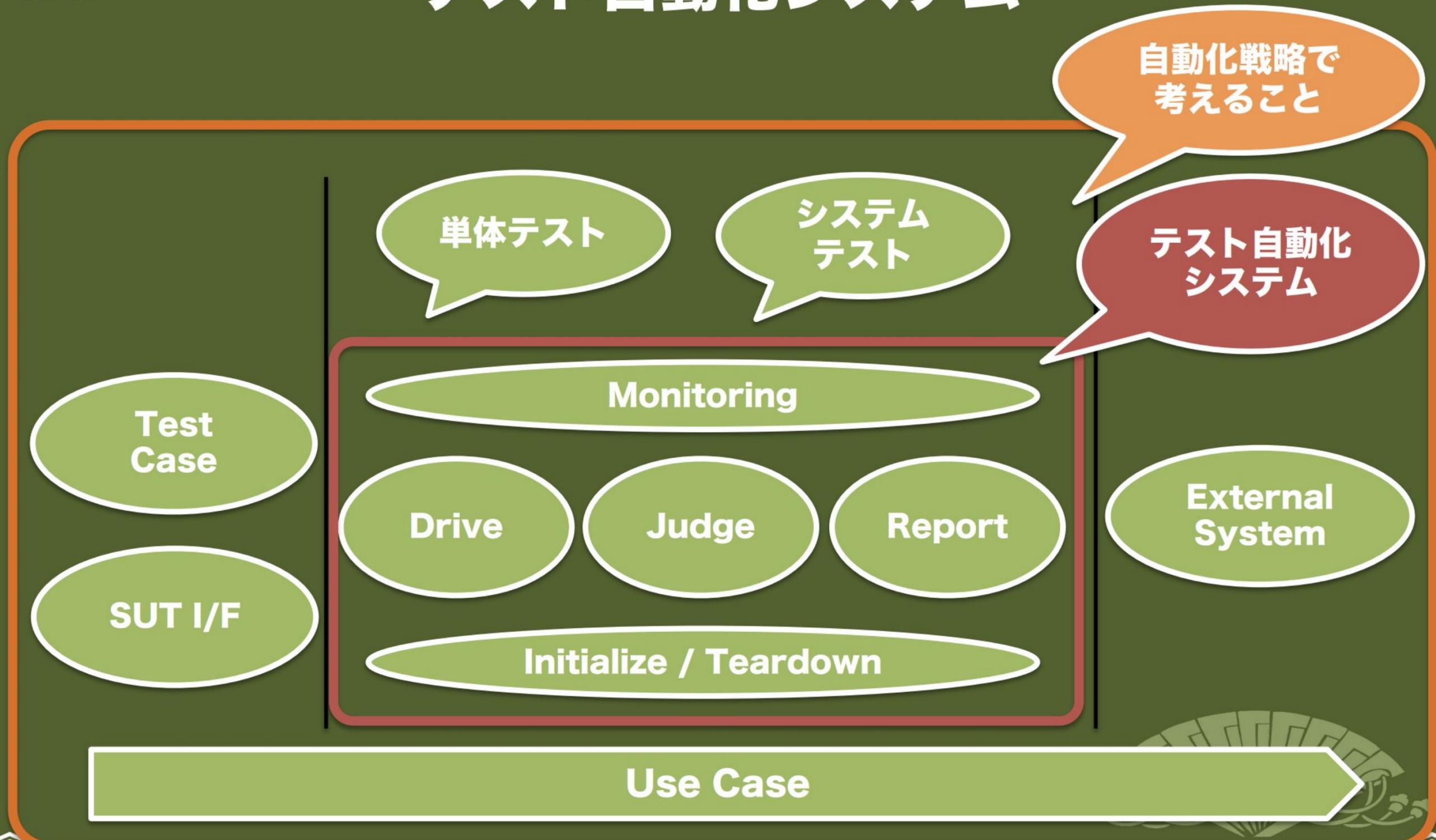


> どうやって構築するの？
いいカンジに構築しましょう

Automation Test.SSF v1.0でいうところの辺



テスト自動化システム



いいカンジに構築しましょう



こればかりは状況(コンテキスト)次第になりますので、
適切なテスト自動化システムになるように構築しましょう。としか言えないです。

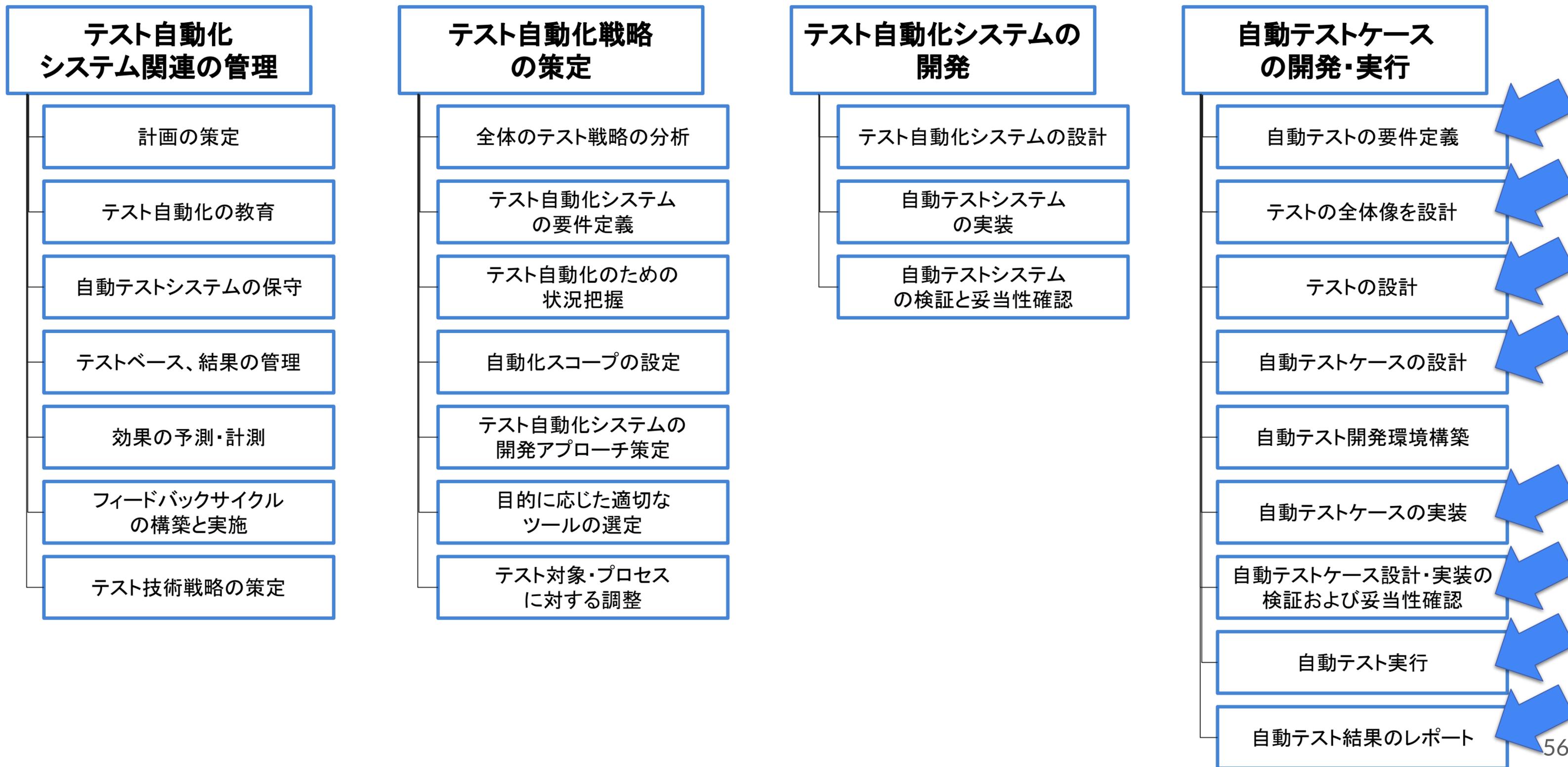
組み込み系ならエミュレーターとかも必要ですよね。
モバイルならデバイスの仮想化とかも必要ですよね。
webならクロスブラウザの仕組みとかも必要ですよね。

個人的に大事なポイントとして以下3つは抑えたい

- ・期間内に目的を達成しやすいのはどういう自動テストシステムなのか
- ・今後も使い続けるサービスとして、組織に取ってもっとも望ましい形はどのような形か
- ・予算内でもっともベターな姿は何か

> 何をテストするの？
自分で考えましょう

Automation Test.SSF v1.0でいうところの辺



テスト自動化に共通する問題(21年前の本に書いてあること)



❖ 非現実的な期待

- 万能を期待される

❖ まずいテストिंगのやり方

- カオスを自動化してもカオスが早くなるだけ

❖ 自動テストは新しい欠陥をたくさん見つけてくれるのではないかという期待

- 動作保証をする目的であって欠陥を抽出する目的ではない

❖ 安全に対する誤った意識

- ずっとpassしている場合、テスト自身が間違っているかもしれない

❖ 自動テストのメンテナンス

- テスト対象が変化すれば自動テストの変化も必要になる

❖ 技術的な問題

- ツールはトレーニングが必要だし、そもそもテスト対象自体がテストしづらいとダメ

❖ 組織の問題

- インフラとして投資が必要であることを組織が理解する必要がある

SWEBOKのソフトウェアテストの定義(再掲)



SWEBOK—SoftWare Engineering Body Of Knowledge—
ソフトウェアエンジニアリング基礎知識体系より引用

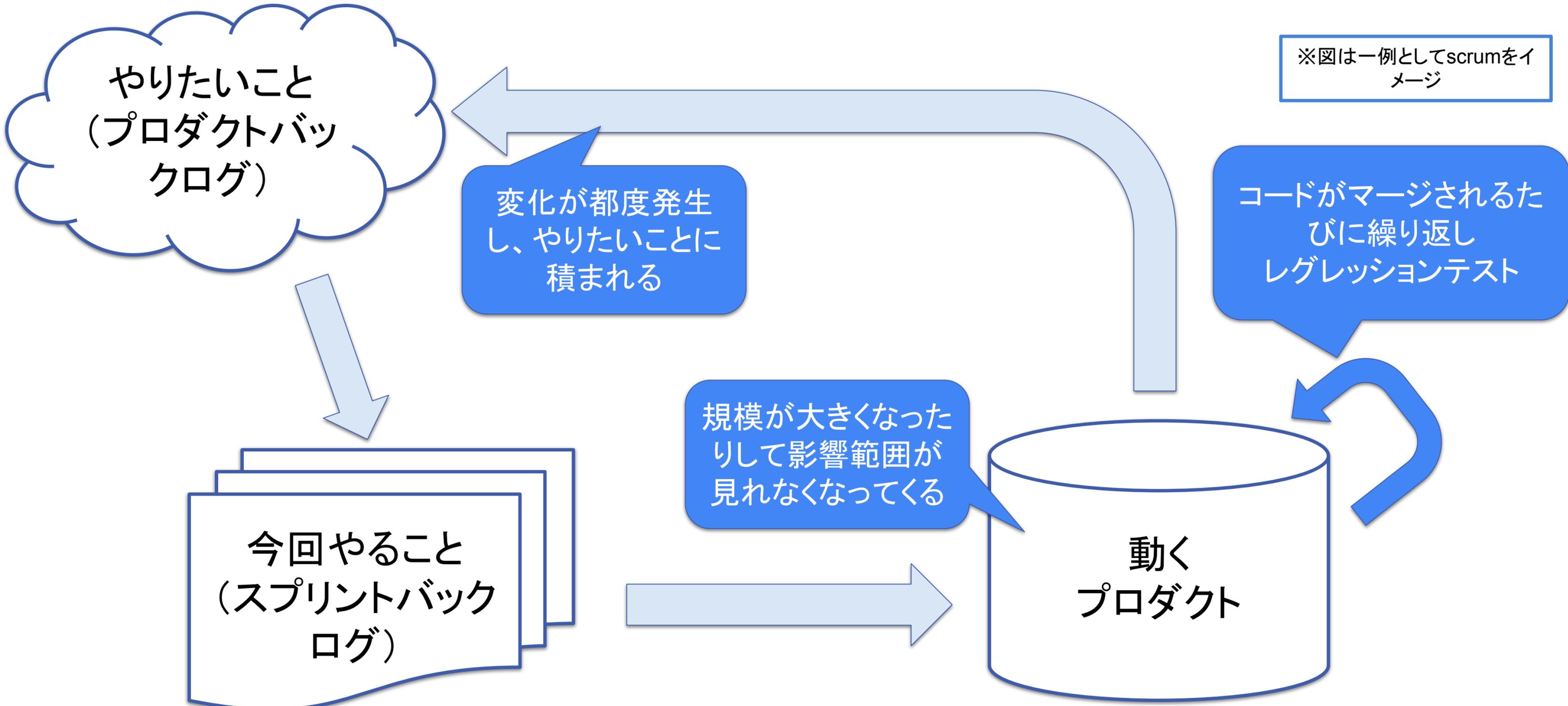
第5章 ソフトウェアテスト

ソフトウェアテストとは、通常は無限に大きいと考えられる
「プログラムの振る舞いの実行領域」から
最適だと考えられる有限な「テストケースの集合」を選定し
所期の通りかどうかを実際に動かして検証すること

アジャイル開発には自動テストがフィットする



※図は一例としてscrumをイメージ



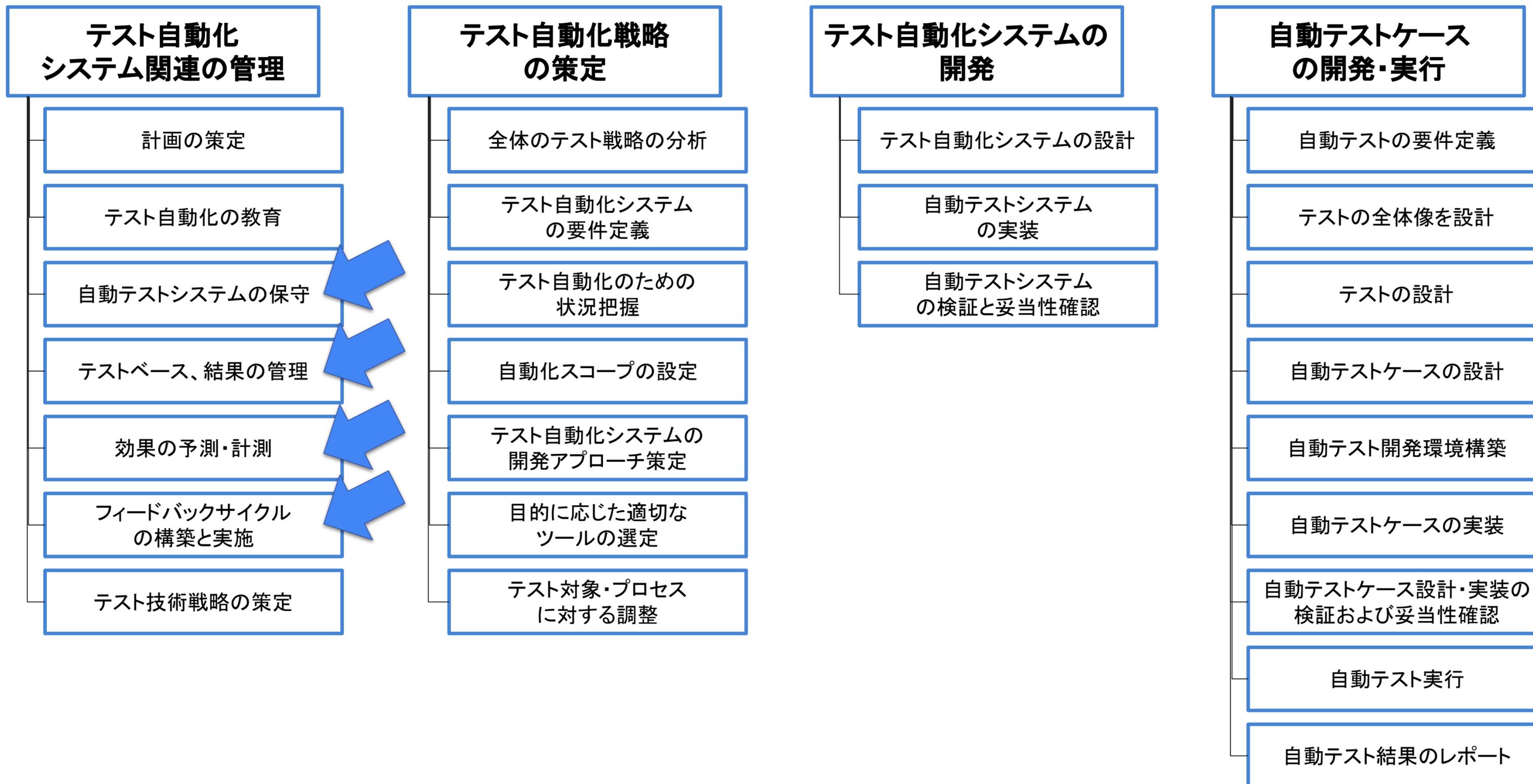
テスト自動化はあくまで手段の1つ
「何を常に知れるようにするか」
「常に安心したいことは何か」
が大事

>いつまでやるの？
ずっとやるんですよ

全体像



Automation Test.SSF v1.0でいうところの辺



テスト自動化に共通する問題(21年前の本に書いてあること)



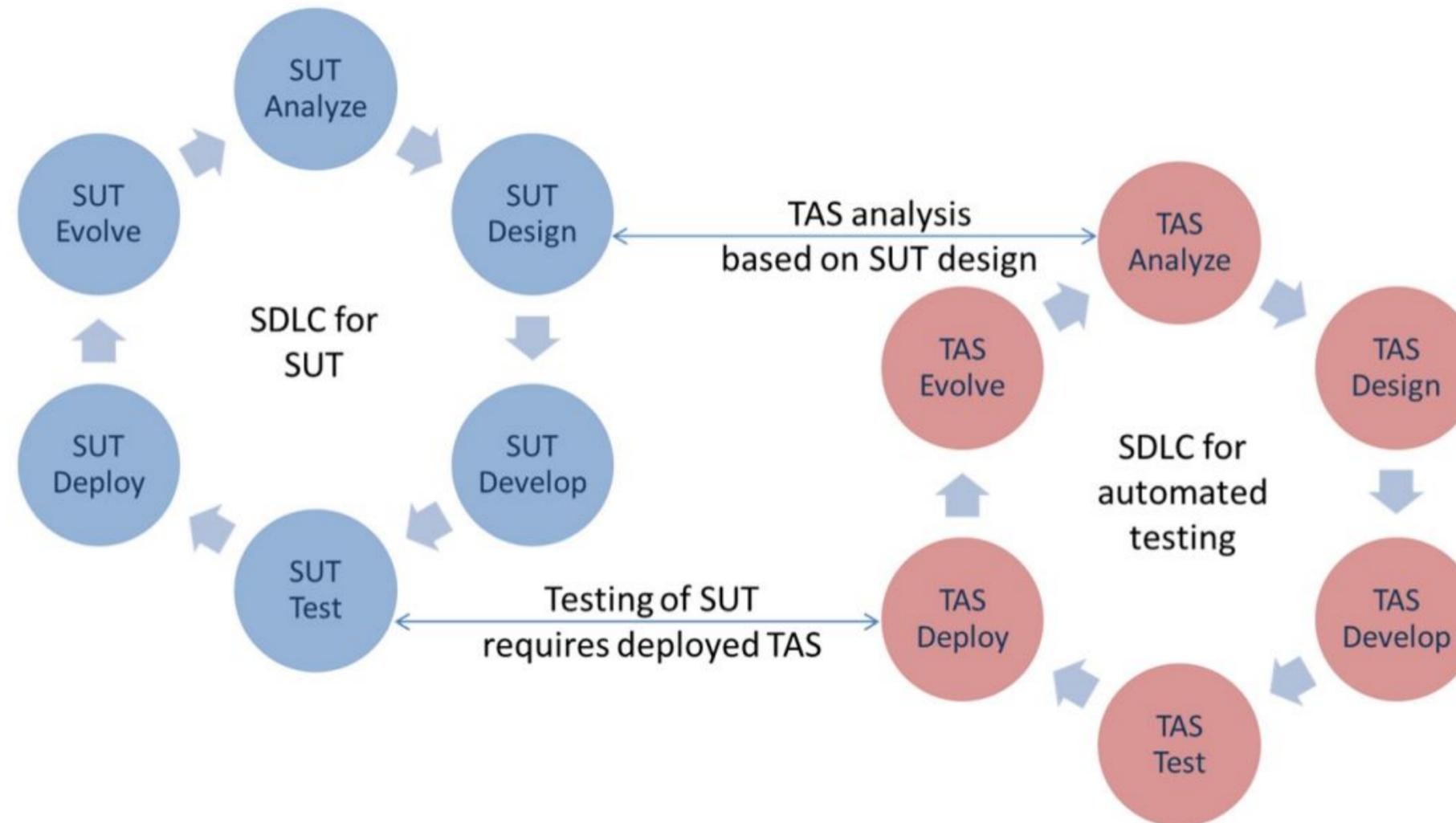
- ❖ 非現実的な期待
 - 万能を期待される
- ❖ まずいテストのやり方
 - カオスを自動化してもカオスが早くなるだけ
- ❖ 自動テストは新しい欠陥をたくさん見つけてくれるのではないかという期待
 - 動作保証をする目的であって欠陥を抽出する目的ではない
- ❖ 安全に対する誤った意識
 - ずっとpassしている場合、テスト自身の間違っているかもしれない
- ❖ 自動テストのメンテナンス
 - テスト対象が変化すれば自動テストの変化も必要になる
- ❖ 技術的な問題
 - ツールはトレーニングが必要だし、そもそもテスト対象自体がテストしづらいとダメ
- ❖ 組織の問題
 - インフラとして投資が必要であることを組織が理解する必要がある

テスト対象に合わせて保守が必要



イテレーティブに開発をしている場合、SUTに合わせてTASも変化する必要がある
これが自動化してもコスト削減にならない理由の一つ。

個人的にはテスト自動化はコスト削減にならないが「常に安心」を提供するサービスと捉えている





Section
03

まとめ

テスト自動化はあくまで手段の1つ
「何を常に知れるようにするか」
「常に安心したいことは何か」
が大事

テスト自動化の大変なところ



- ❖ いつ構築するの？
 - テスト対象に合わせてみましょう
- ❖ どうやって構築するの？
 - いいカンジに構築しましょう
- ❖ 何をテストするの？
 - 自分で考えましょう
- ❖ いつまでやるの？
 - ずっとやるんですよ

テストは無限から
最適だと考えられる有限
を導き出すテクニック



Section
04

宣伝

freeeは名古屋拠点立ち上げメンバー募集中です！



私一応QAアニキ(QA Tech Lead)というロールなんですが、
弊社QAメンバーはゆもつよさんを筆頭に強い人多すぎて3～4番手くらいです。
そんな凄腕QAたちと働いてみたいエンジニアの方は是非チェックしてください！

<https://freecommunity.force.com/jobs/s/detail/a4l2r000000CaRWAA0>

<https://www.wantedly.com/projects/497238>

働きやすくてオススメです

<https://jobs.freee.co.jp/benefits/>

名古屋拠点エンジニアmoaiがイベント企画してます



freeの雰囲気などを知りたい方は是非弊社エンジニアのmoaiが主宰しているイベントがあるようなので、是非こちらもチェックしてください！

<https://misokatsu-web.connpass.com/event/191753/>

スモールビジネスを、
世界の主役に。





Section
05

Q&A