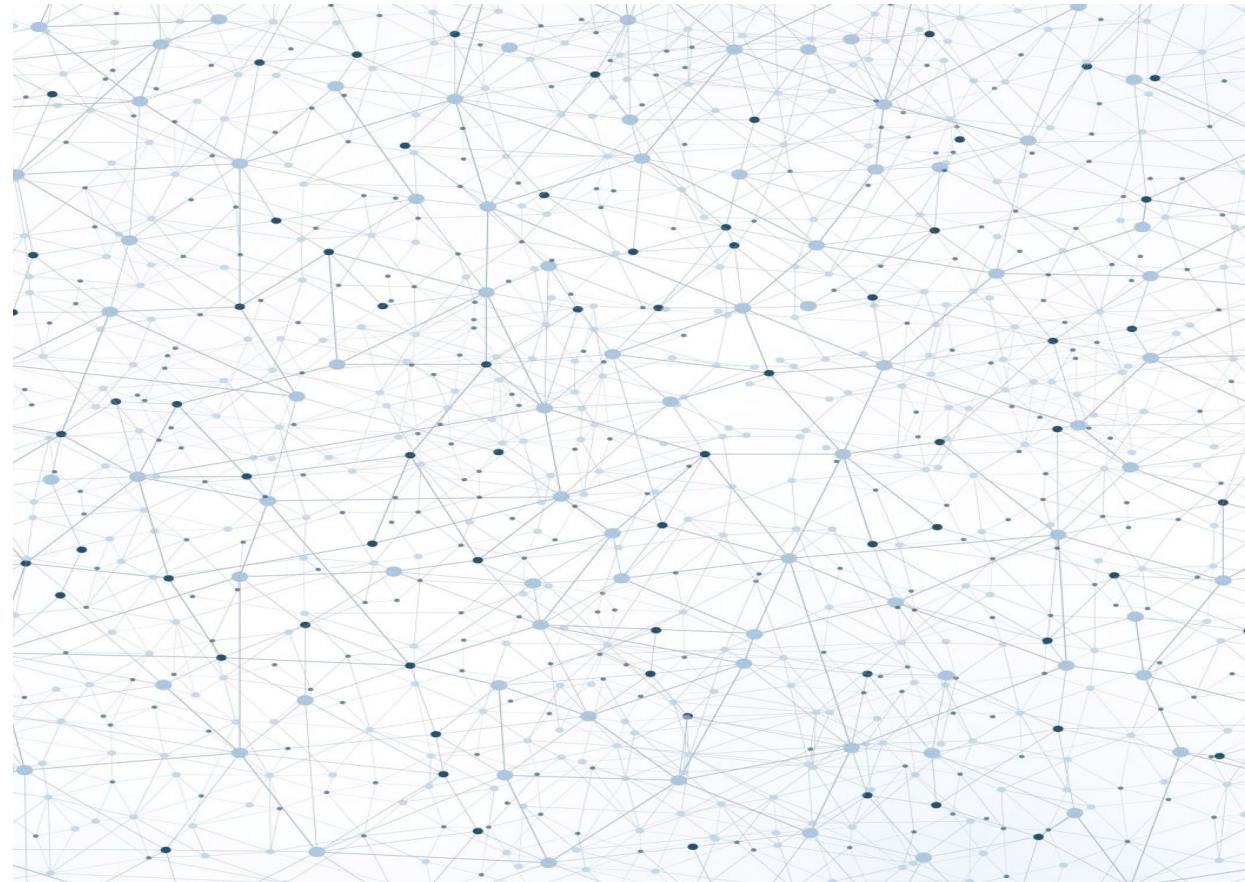

ソフトウェアテスト技術振興協会 (ASTER)調査研究事業の活動紹介

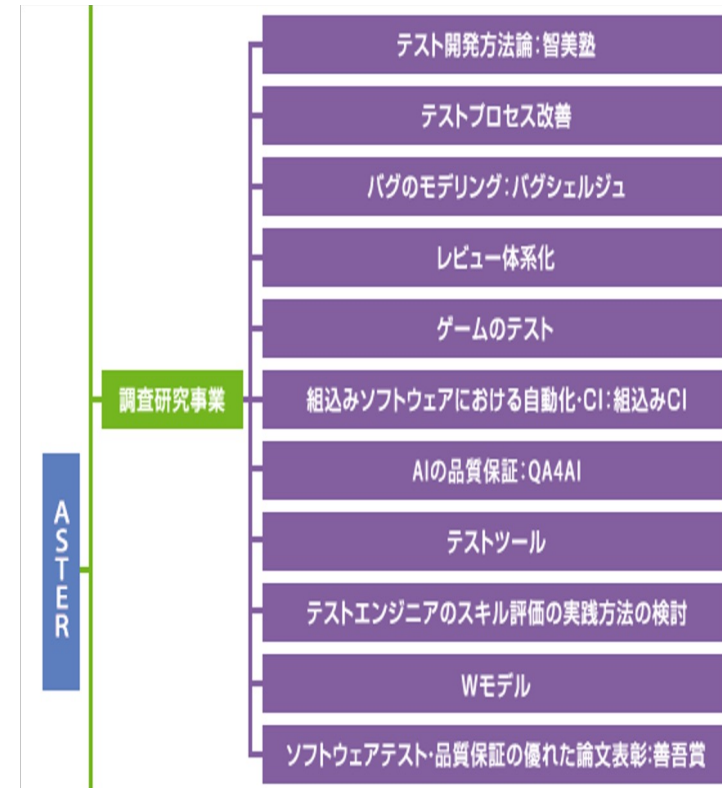
2026/3/20

JaSST'26 Tokyo@東京ビッグサイト



ASTER事業と調査研究事業

- NPO法人ソフトウェアテスト技術振興協会(ASTER)は、ソフトウェア品質向上に関する技術力の向上や普及活動を行うことで高度情報化社会に寄与するために以下の事業を行っています。
 - ソフトウェアテストシンポジウム (JaSST) などの普及振興事業
 - テスト技術者資格認定Japan Software Testing Qualifications Board (JSTQB) 運営などの資格認定事業
 - テストに関連する技術開発や技術力向上のための**調査研究事業**
 - アジア圏との連携 (AsiaSTA) や国際的なテスト・品質動向の調査支援などを行う国際連携事業
 - 国内外各地域における勉強会の支援や講師派遣などの教育事業

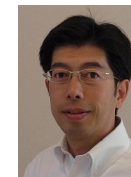


本セッションの内容

- 本セッションで紹介する各活動（各5分間程度）
各テーマに分かれて活動しています。今回はその活動を紹介します。
 1. ソフトウェアテスト調査研究会(Test-SIG) [増田]
 2. 機械学習システムのQA(QA4AI) [徳]
 3. テスト開発方法論(智美塾) [來山]
 4. バグモデリング(バグシエルジュ) [嬉野、伊藤]
 5. ゲームのテスト [勅使川原]
 6. 組込みCI [森、林]
 7. テストツール [須原]
 8. レビュー [安達]
- ミニチュートリアル（20分間程度）
 - 大規模言語モデル(LLM)によるテスト技術のための基礎知識 [武田]

ソフトウェア テスト調査研 究会(Test- SIG)の紹介

- 2025年4月から活動を開始。
- ソフトウェアテストやソフトウェア品質に関する研究を行い論文等で発表する。
- 月1回程度のオンライン打ち合わせ。
- メンバーそれぞれで年1本程度の論文投稿を目指す。
 - 現在メンバー：増田聡(都市大)、武田友宏(富山大客員教授)、加藤敏春(筑波大学ビジネスサイエンス系研究生)
- 2025年度活動
 - 国際学会に論文が2件採択された
 - ICICT2026、“AI Agent Case Study Generating High-Level Test Cases from Requirements”
 - ICSE2026-SERS, “Testing Research Software using International Standards and Practices”
 - ジャーナル論文1件投稿





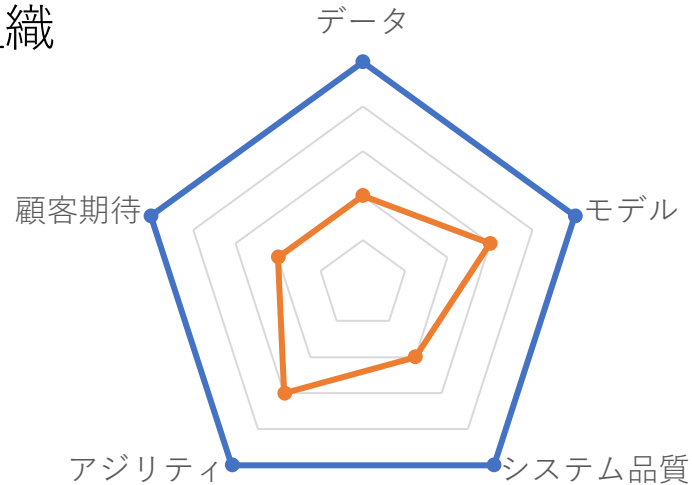
Consortium of
Quality Assurance for
Artificial-Intelligence-based
Products and services

AIプロダクト品質保証コンソーシアム

<https://www.qa4ai.jp>

・AIプロダクトの品質保証における産業界のベストプラクティスやガイドライン発行を2018年より開始

・データ・モデル・システム品質・プロセスアジリティ・顧客の期待の5軸に対し、プロダクトの進展に合わせてバランスよく成長すべきという考え方を基本とし、エンジニア思考を中心とした具体的なガイドラインの策定をする組織



- ① Data Integrity (データの量、質、管理 など)
- ② Model Robustness (性能尺度、多様な検証、監視 など)
- ③ System Quality (提供価値、インシデント許容度、結合度 など)
- ④ Process Agility (データ収集反映、リリースサイクル、構成管理、省力 など)
- ⑤ Customer Expectation (データの質や量への理解、説明性や原因追及の期待値、実運用での協力度 など)

・これまでの活動：機械学習活用での共通技術（テスト方法、説明性など）と、ドメイン領域（自動運転、産業用、コンテンツ生成、Voice User Interface、AI-OCR、対話型生成AI）で、品質特性やプラクティスのガイドを発行

・昨今は生成AIの急速な普及と機能強化の流れを受けて、ドメインごとの検討は停止し方針を転換。生成AIを機能に生かす場合の品証だけでなく、開発での生成AI適用や、開発者の教育やプロセスに焦点

・活動は副運営委員・ASTER会員の徳隆宏を中心に活動しており、参加希望者は問題意識やプラクティス共有の方向性について準備し運営に連絡ください。

智美塾のご紹介

智美塾訓

1. 塾生はテスト開発方法論を目指し、自ら技術獲得に励む事
2. 塾生は現場を尊び、地に足の着いた技術論を行う事
3. 塾生は品質向上という最終目的を常に意識した活動を行う事



塾長（吉澤さん）

智美塾って？

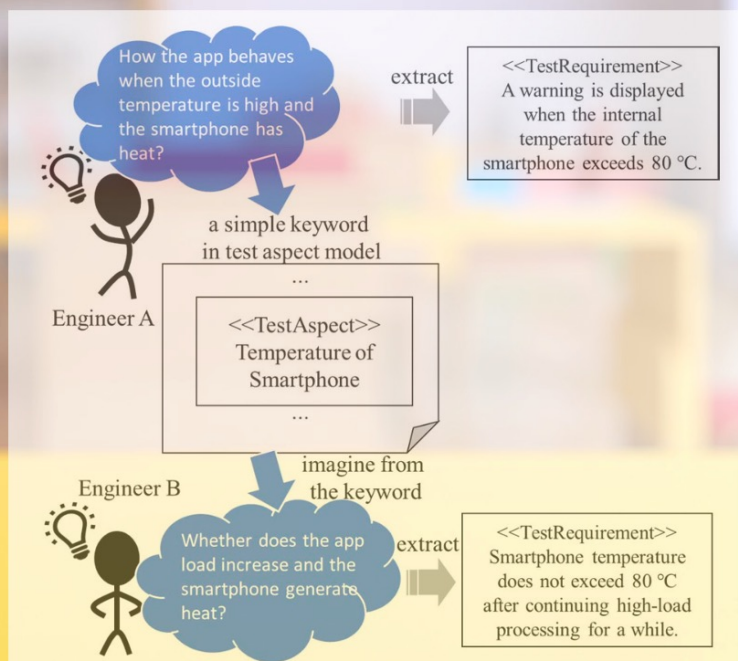
よいテスト設計をするためのアレコレを調査したり開発したりする研究会です！

テストや品質に関心を持つエンジニアが集ってテスト設計に関わる技術や関心事を話し合っています。

年内いつでも、参加可能です！

開催形式	: オンライン
開催頻度	: 3ヶ月に2回程度
現在活動中のメンバー	: 6名

智美塾のご紹介



これまでの活動

2009	JaSST Tokyoのクロージングパネル テスト技法からテストメソトロジへの進化を目指して をきっかけに、テスト開発方法論の研究を行う智美塾が始まる
2010-2015	テスト要求分析 - テストアーキテクチャの議論 JaSST Tokyoで 毎年研究成果発表を行う
2017-2020	InSTAに論文投稿・発表を行う UTP(UML Testing Profile)に対して、テスト担当者が抱く「気がかり：Test Aspects」の概念を提案 2018：「Proposal for Enhancing UTP2 with Test Aspects」 2019：「Designing Fulfilling Test Cases with Test Aspect Model」 2020：「Techniques to Have a Common Understanding of Test Aspects among Test Team Members」
2022-2024	テスト観点とは何か？Test Aspectsとの違いは何か？を考える 公表された文献に掲載の「テスト観点」と品質特性との紐付けを試み SQuaRE の理解を深める
2025-	AIプロダクト品質保証ガイドライン(AIプロダクト品質保証コンソーシアム 編)の読書会を行う

智美塾 メンバー 募集中

これからやってみたいこと

過去の論文をもっと
深掘ってみたいな～

抜けのないテスト設計を
目指すためにどうすれば
いいか考えたい！

過去の論文を和文に
したいな～

テストアーキテクチャにつ
いてもっと議論したい！

先生はいないよ。
みんなで考える場だよ。
全員が発言するよ。
お互いに気づきをシェアして
理解を深める場を
みんなでつくるのが智美塾だよ。



塾長（吉澤さん）

お申し込み : aster-stmjk-req@npoaster.jp
詳しくはコチラ : [智美塾のご案内](#)



過去の資料・論文などのリンク

[JaSST 2011](#) [JaSST 2012](#)

[Proposal for Enhancing UTP2 with Test Aspects](#)

[Designing Fulfilling Test Cases with Test Aspect Model](#)

バグシエルジユ

Bugcierge



主なメンバー



嬉野 綾

バグシエルジユ / SReEE(レビュー研究会)



吉澤 智美

NEC



湯川 健

ナレッジワーク



渡邊 大輔

ウイングアーク 1 s t



永田 敦

三菱電機



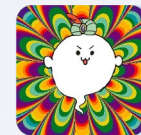
安楽 啓之

インフォテック



伊藤 瑞紀

Works Human Intelligence



山下 友輔

testingOsaka

and more...

バグシエルジュ：これまでの活動

バグを分析して、対策しても
また同じようなバグが出る...

チェック体制を強化しました！
で充分なのか...？

JaSST 2012 Tokyo – Project Fabre

Project Fabre presents
欠陥マスター情報構築ワークショップ

—「悪さの知識」伝承によるバグ予防の為のバグ情報マスター化実践—



Nobuhiro Hosokawa, Yasuharu Nishi,
Aya Ureshino, Makoto Nonaka, Yukiko Hara

JaSST 2013 Tokyo – Project Fabre

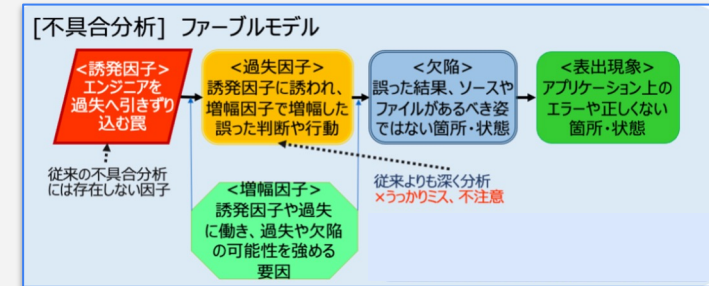
過失に着目した欠陥のモデリング

—バグ分析はなぜうまくいかないのか？—



Nobuhiro Hosokawa, Yasuharu Nishi,
Aya Ureshino, Makoto Nonaka, Yukiko Hara

2012 [JaSST](#)
2013 [JaSST](#)
2018 [ISSRE](#)



『人間の過失に注目してバグ分析を実施』

人の陥りやすい罠を見つけて

不具合を未然防止

- ひとりでも多くの方が安心できる社会を -

10. バグ分析ができない事例の特徴

多くの場合、下記ができていない事例が多い
→できるようにすれば、解決へ近づける

- バグを楽しむ
- バグを表現する
- バグを比較する
- バグを移転する
- バグを蓄積する
- バグを資産とみなす

過失に 着目！



バグシエルジュ：現在の活動（月1で活動しています）

論文の翻訳

Trap-based Software Review: How will Engineers Embed Bugs due to Cognitive Error?

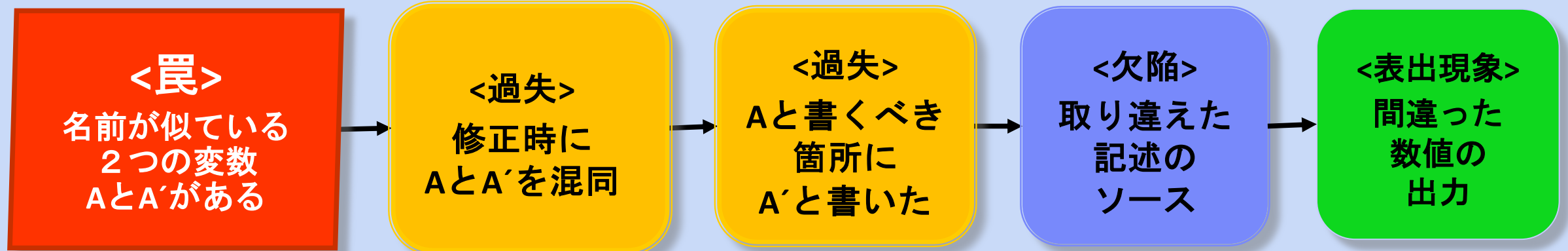
トラップベースのソフトウェアレビュー：エンジニアは認知エラーによるバグをどのように埋め込むのか？

- ・ 国際会議 ISSRE2018

Best Disruptive Idea Award（最優秀破壊的アイデア賞）

- ・ **もっとみんなに見てもらって、役立ててほしい**
- ・ 翻訳版をASTERのサイトで公開予定です

『そっくりさん』の罫（トラップモデルを用いた罫のご紹介）



とりちがえたまま開発が進み、数値の不具合が発生！！¹

新たな「罫」の発見

- ・ 実際のバグから「トラップモデル」を用いて「罫」を発見する
- ・ 罫のパターンを増やして、レビューで活かせる知識にする

「トラップモデル」を究める

- ・ バグ情報を抽象化する「トラップモデル」をわかりやすく解説
- ・ 論文の補足&解説
- ・ 現場のバグ分析サポート

『人間の過失に注目してバグ分析を実施』

人の陥りやすい罫を見つけて

不具合を未然防止

一緒にやりませんか？

- ・ バグ分析に興味がある
- ・ レビューをもっと良くしたい
- ・ 分析したいバグを持っている

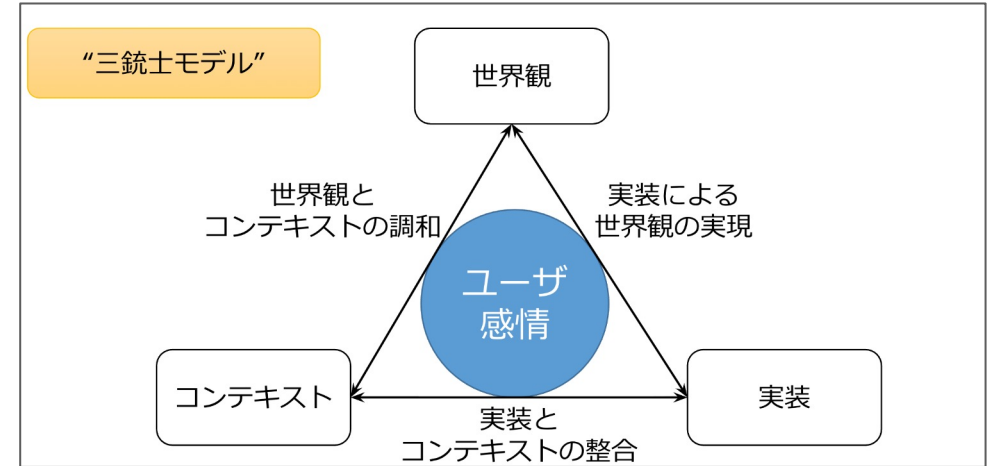
ゲームテスト研究会の紹介

● 目的

- 複雑・大規模なゲームテストを楽に（効率的に）したい

● 活動内容

- 三銃士モデルの実務応用
- JSTQBゲームテストシラバスの輪読
- JaSST発表
- 書籍の執筆



ゲームテストのためのテスト観点モデル「三銃士モデル」

JaSST登壇実績

- JaSST'16 Tokyo
「テスト設計；ゲームテストの世界」
- JaSST'17 Tokyo
「続：テスト設計；ゲームテストの世界」
- JaSST'17 Kyushu
「ゼロから始める "テスト設計" の書 (ゲームテストの世界)」
- JaSST'18 Tokyo
「三銃士モデルではじめる、テスト分析ワークショップ」
- JaSST'19 Tokyo
「三銃士モデルを適応してみた！ -Next Generation 次世代の挑戦-」
- JaSST'21 Tokyo
「ゲームテストにおけるユーザー感情という「想いは不滅」」
- JaSST'22 Tokyo
「ゲーム業界に転生してゲームテスト大百科を作ってみた」
- JaSST'23 Tokyo
「『ゼロからはじめるゲームテスト』その執筆はLvゼロからはじまった」

執筆書籍



『ゼロからは始めるゲームテスト
-壁抜けしたら無限ガチャで最強モードな件？』

- きっかけ
 - 「ゲームテスト」の書籍が少ない
 - 積み重ねた知見をまとめたい
- 執筆の流れ
 - テーマ・ターゲットを決定
 - 全体構成・ネタ出し
 - 分担して執筆
 - 編集者の方からのFB・修正
 - 出版

活動メンバー募集中！

所属はゲーム・非ゲーム問わず、
ゲームQAに興味ある方歓迎！

- 活動頻度
 - 月1回程度（オンライン）
- 今後のテーマ（予定）
 - 三銃士モデル×AIの活用

連絡先



ASTERへのメール

OR

daisuke.teshigawara@gree.net

勅使川原まで

組込みCI研究W/G活動紹介

「組込みCI研究W/G」 JaSST'26 Tokyo(2026/3/20@東京)
ご興味ある方は「aster-embci@npoaster.jp」まで。ご連絡をお待ちしております。

①組込みCI研究W/Gの経緯

テスト自動化のユーザーイベントで、組込み製品のテストも、非組込み分野のようにCIが使えないの？と盛り上がった有志達により、本W/Gが2018年に発足しました。議論に参加したい方は分野によらず、参加をお待ちしております。

②W/G発足当初の問題

- やってきた仕事の内容や製品の違いが大きく、話が噛み合わなかった。
- 皆の興味の対象が違いすぎ、W/Gの方向性が定まらなかった。

③発足当初の対策

「システムテスト自動化標準ガイド」「Googleのソフトウェアエンジニアリング」を輪読をして、皆の知識レベルを揃える。

④組込みならではの検討課題

- **ハードに近い部分のテスト自動化**の検討。
- 製品のInput/Outputのデータだけでなく、**電圧、電流などのハードの特性を理解し、自動化装置の環境構築**。
- 組込み製品のテスト自動化に向く内容の検討。
- **ハードへの書き込みの自動化方法**の確立。
- 一部を除き**インターフェースが統一されていない**。
- **派生開発が多く**、複数プロジェクトでソフトが再利用される。
- ECUなど組込み製品にサーバサイドで進んでいる**仮想化技術の適用方法**の検討。

⑤25年度活動

- 「ソフトウェアテストをカイゼンする50のアイデア」を輪講
- 組込みソフトウェアテストの改善アイデアを検討
- JaSST東海のSIGにて上記アイデアのワークショップを開催

組込みCI研究W/G活動紹介

「組込みCI研究W/G」 JaSST'26 Tokyo(2026/3/20@東京)
ご興味ある方は「aster-embci@npoaster.jp」まで。ご連絡をお待ちしております。

JaSST'25 Tokai でのSIG内容

組込みソフトウェアテスト：次の一歩につながるアイデア共有会



内容

- ・組込み開発でのテストでよくあるテーマでのディスカッション
- ・どのような改善の一歩ができるかを相談

「組込みSWテストの改善アイデア集」

目的

組込みSWの自動テストを実現する際の困り事を解決するための**改善アイデアやヒントを得ること**

＜組込みSWでのシステムテスト自動化改善のアイデア例＞
まずはテスト対象のハード入出力信号を身近な装置で模擬できないか？を考えよう

＜主な利点＞

新しい装置や高精度な装置を導入すると対応工数が増大
使い慣れている装置が使えるのかを考えることで、スモールスタートを可能にする。

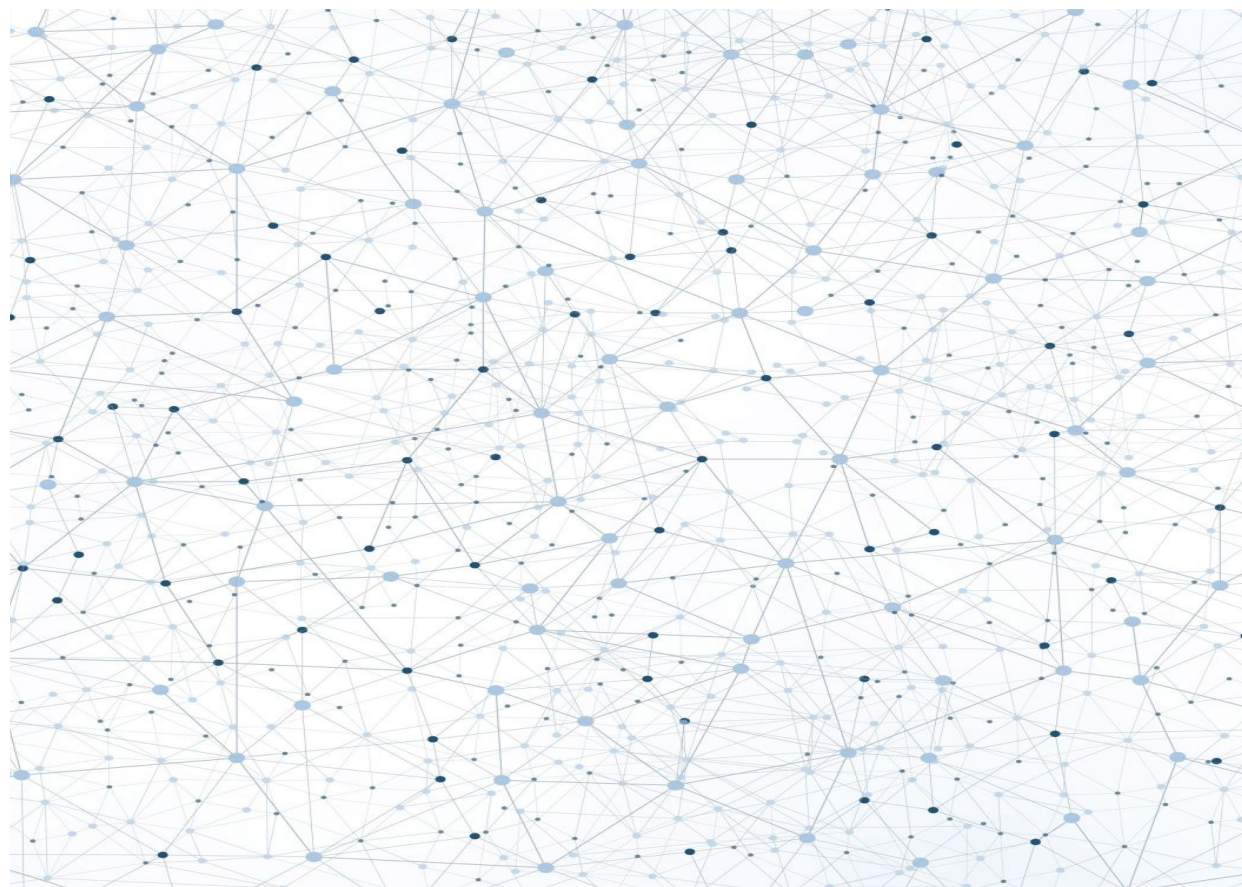
＜それを機能させる方法＞

- ・模擬できる部分を見極め、最低限のテスト「例：HWとSWの連携のみ」を実行できるようにする。
- ・システム・ハードのテスト分担を考えて、ソフトウェアチームでのテストの範囲を決める。

テストツールWG

2026/3/20

JaSST'26 Tokyo F2セッション



テストツールWGとは

- テストツールまるわかりガイドを編集、発行するWGです。企業間の垣根を越えて多様なツールを知っていただくために、国内のテストツールベンダ／販社各社に協力を呼びかけ、国内で入手可能な各種テストツールの基本情報を得られるようにしています。
 - 2012年にv1.0発行
 - 2020年にv2.0発行

テストツールまるわかりガイドの構成

NPO 法人 ソフトウェアテスト技術振興協会

テストツールまるわかり ガイド(ガイド部)

Version 2.0.0

2020/9/30

	A	B	C
1	ツール名	何のためのツールか一言で	ツール分類
2	ALM Octane	エンタープライズアジャイル管理ツール	テストケース管理ツール, テスト結果/レポートツール, 要件管理ツール, インシデント管理ツール, レビュー支援ツール, 原因結果グラフツール, テスト設計支援ツール, テスト自動実行支援ツール
3	ALM/QualityCenter	アプリケーションライフサイクル管理ツール	テストケース管理ツール, テスト結果/レポートツール, 要件管理ツール, インシデント管理ツール, レビュー支援ツール, 原因結果グラフツール, テスト設計支援ツール, テスト自動実行支援ツール
4	Application Diagnostics	アプリケーションパフォーマンス分析ツール	動的解析ツール
5	Autify	アジャイルチームのためのQA自動化プラットフォーム	テストケース管理ツール, テスト結果/レポートツール, キャプチャ/リプレイツール

テストツールまるわかりガイドの構成

- 「ガイド部」と「ツール部」から構成されています
- ガイド部は変化が少ない解説部分、ツール部はどんどん追加する部分として分けています

現状と課題

- 2020年の発行以降、頻繁にアップデートできているかという
とそうではないです
- さらに各種ツールのリリースもどんどん行われており、常時
更新できる仕組みが必要だと考えています
- 一緒に考えてくださる方、募集中！



これまでの活動と現在

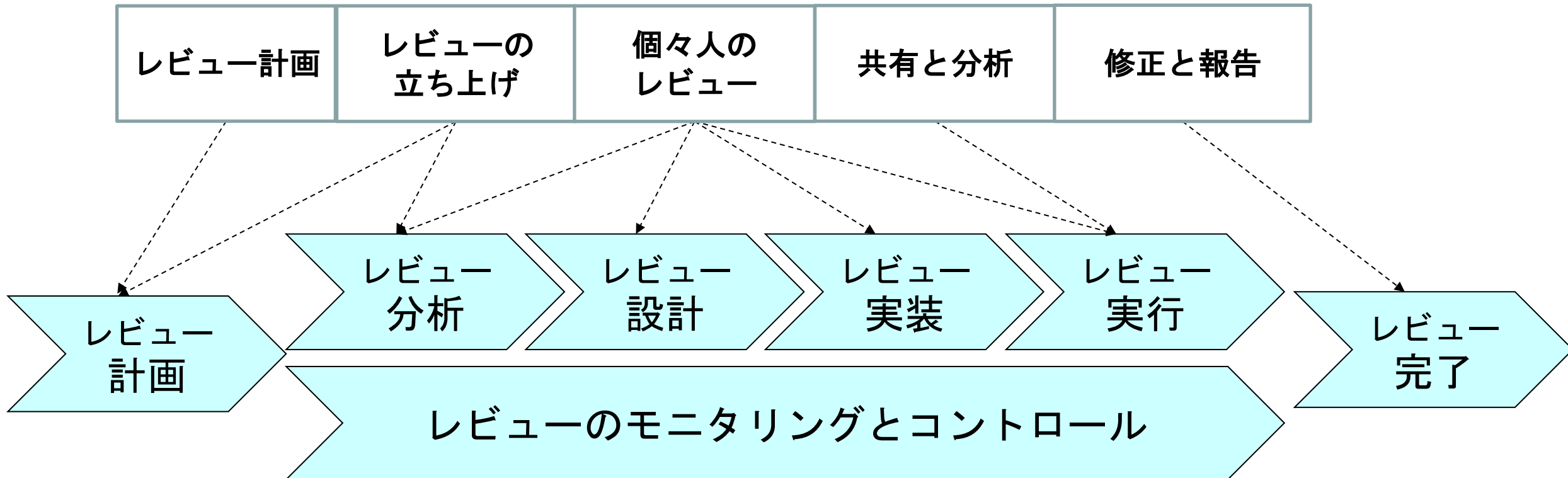
- 2017年～1回/月程度研究会開催
- JaSST Reviewの企画と運営
- レビューに関する記事を連載中
 - <https://sscripts.com/writer/sreee/>

ARTICLE このスクリプターの執筆記事

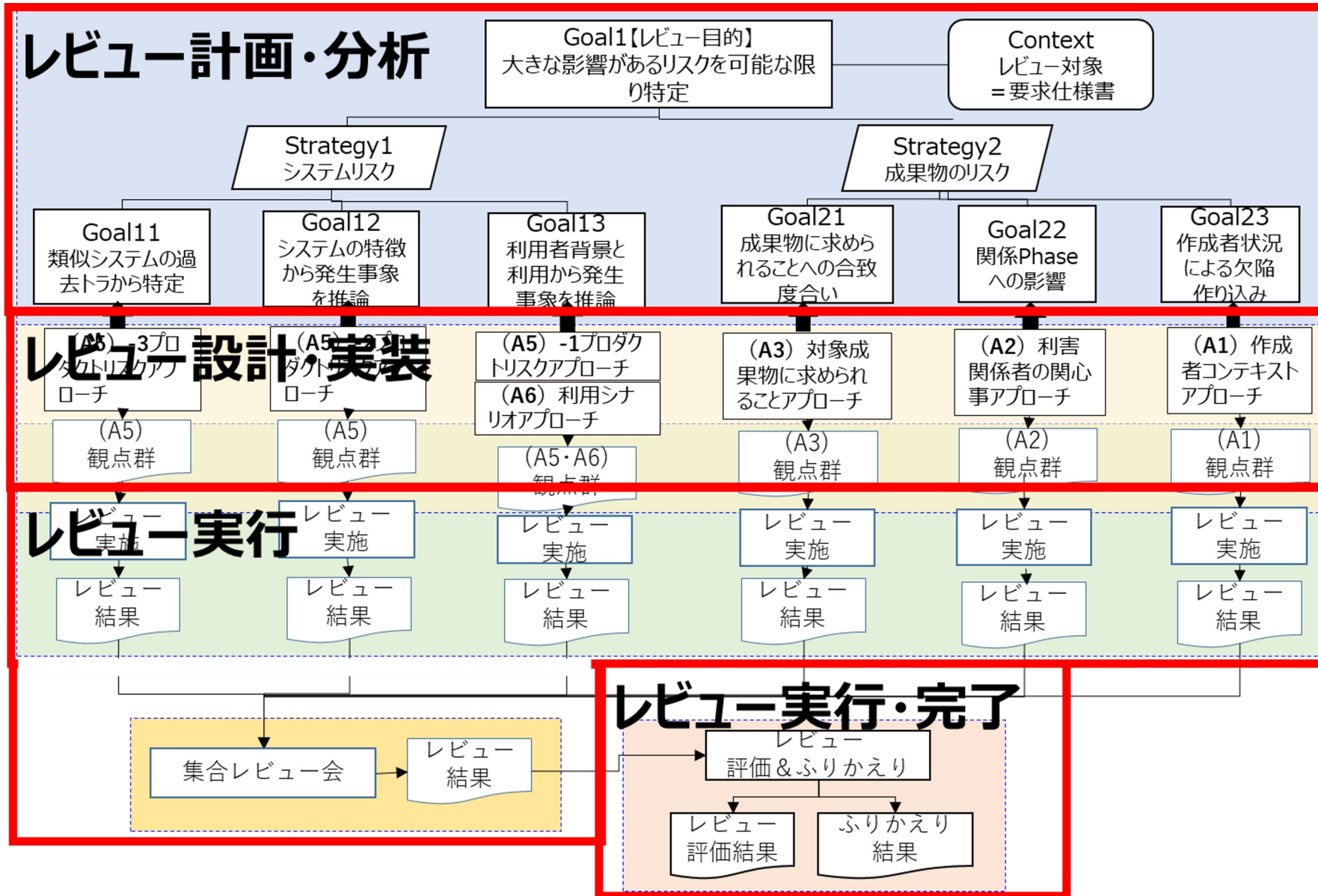
- 
【第4回】[#5]レビューの終わり方～レビュー評価・ふりかえり(2/2)|ソフトウェアレビューをエンジニアリングっぽく捉える
 <2025/05/07>
- 
【第3回】[#5]レビューの終わり方～レビュー評価・ふりかえり(1/2)|ソフトウェアレビューをエンジニアリングっぽく捉える
 <2025/04/23>
- 
【第2回】[#1]レビューとは?|ソフトウェアレビューをエンジニアリングっぽく語ってみる
 <2025/03/19>
- 
【第1回】[#0]イントロダクション|ソフトウェアレビューをエンジニアリングっぽく語ってみる
 <2025/02/05>

検討中の「レビュープロセス」

テストプロセスを参考にしたレビュープロセスの分解



レビュープロセス実践ワーク & 解説



2026/3/20 JaSST'26東京
ASTER R&Dセッション

大規模言語モデル(LLM)による テスト技術のための基礎知識

武田友宏 Ph.D. (工学博士)

富山大学: 客員教授

Amazon Web Service: Senior AIML Specialist

Agenda

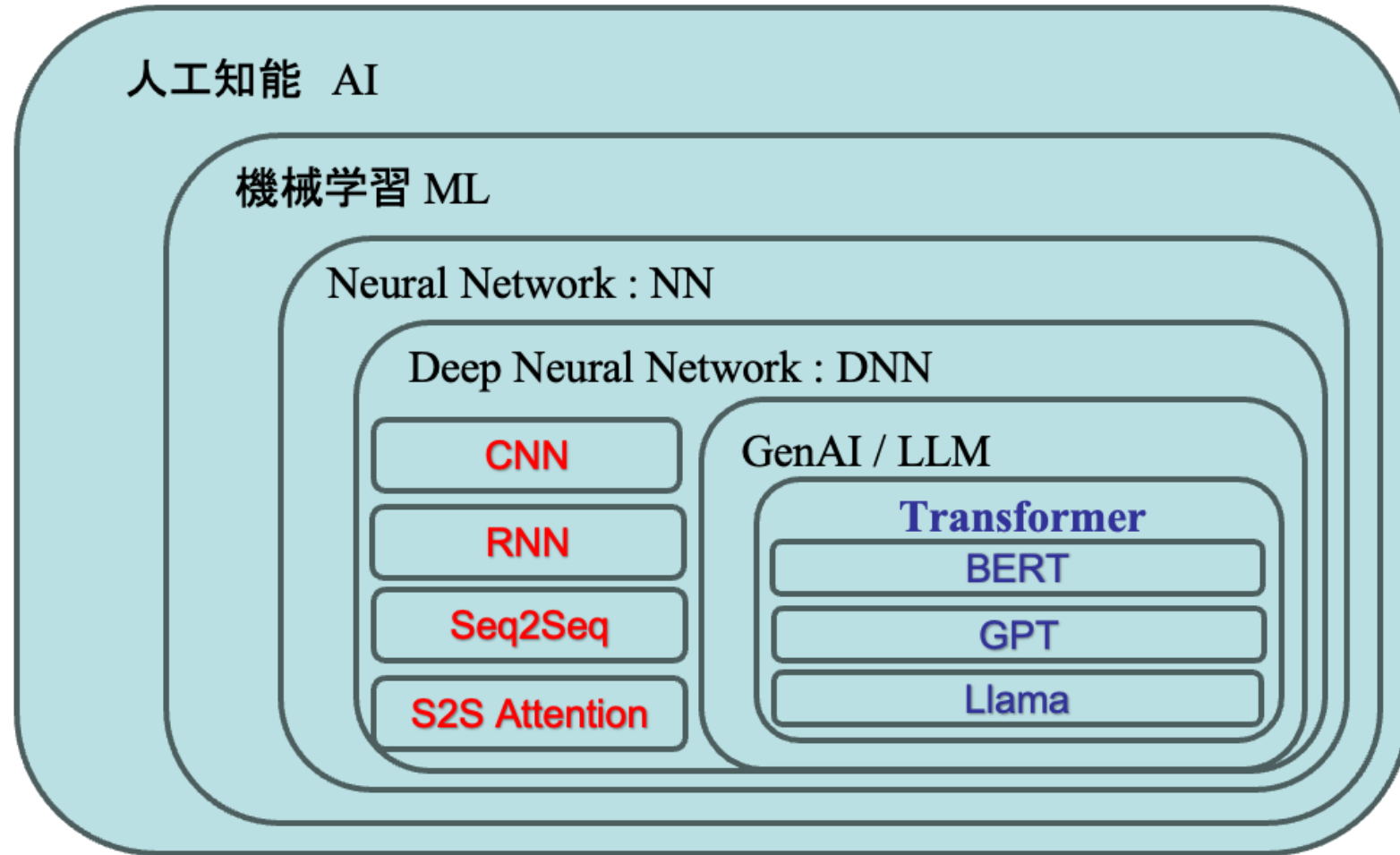
1. 生成AI進化の歴史

1. 生成AIの誕生
2. Neural Networkの学習とは？
3. Transformerの学習とは？
4. 進化①②③④⑤

2. 生成AI時代におけるテストエンジニアリング

1. テスト対象と観点の変化
2. Transformer

1. 生成AI進化の歴史



AIの技術要素Map

1.1. 生成AIの誕生

2017

Multi Layer

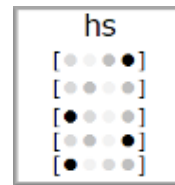
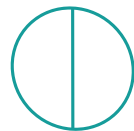
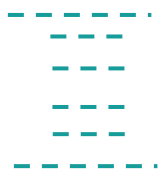
FeedBack

Gate

Encoder
Decoder

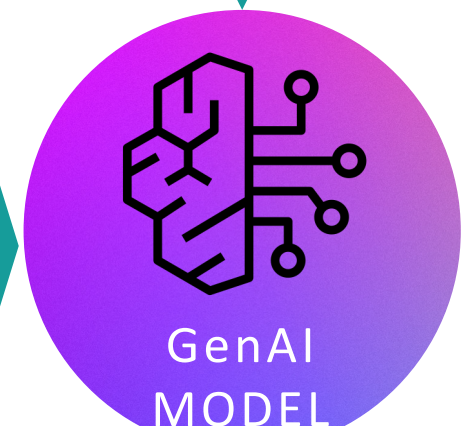
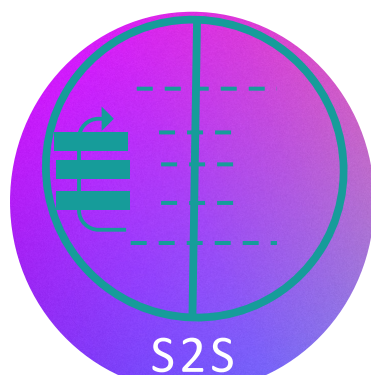
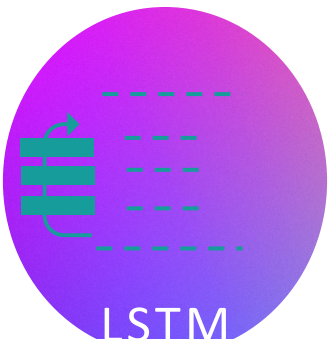
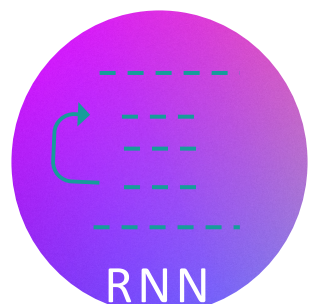
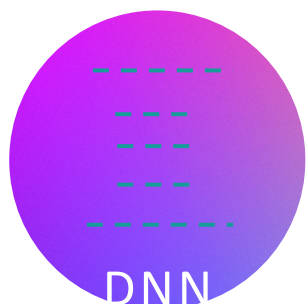
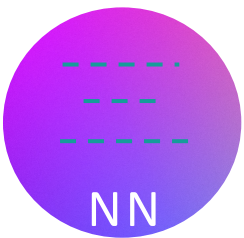
Attention

Position Encoding



$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d}}\right)$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d}}\right)$$



シワ3つ
3 Layer

シワ沢山
Multi Layer

逆流
Feedback

忘却
Gate(Forget)

右脳左脳
Separated

関連性
Attention

関連性だけで
理解する
Attention Only

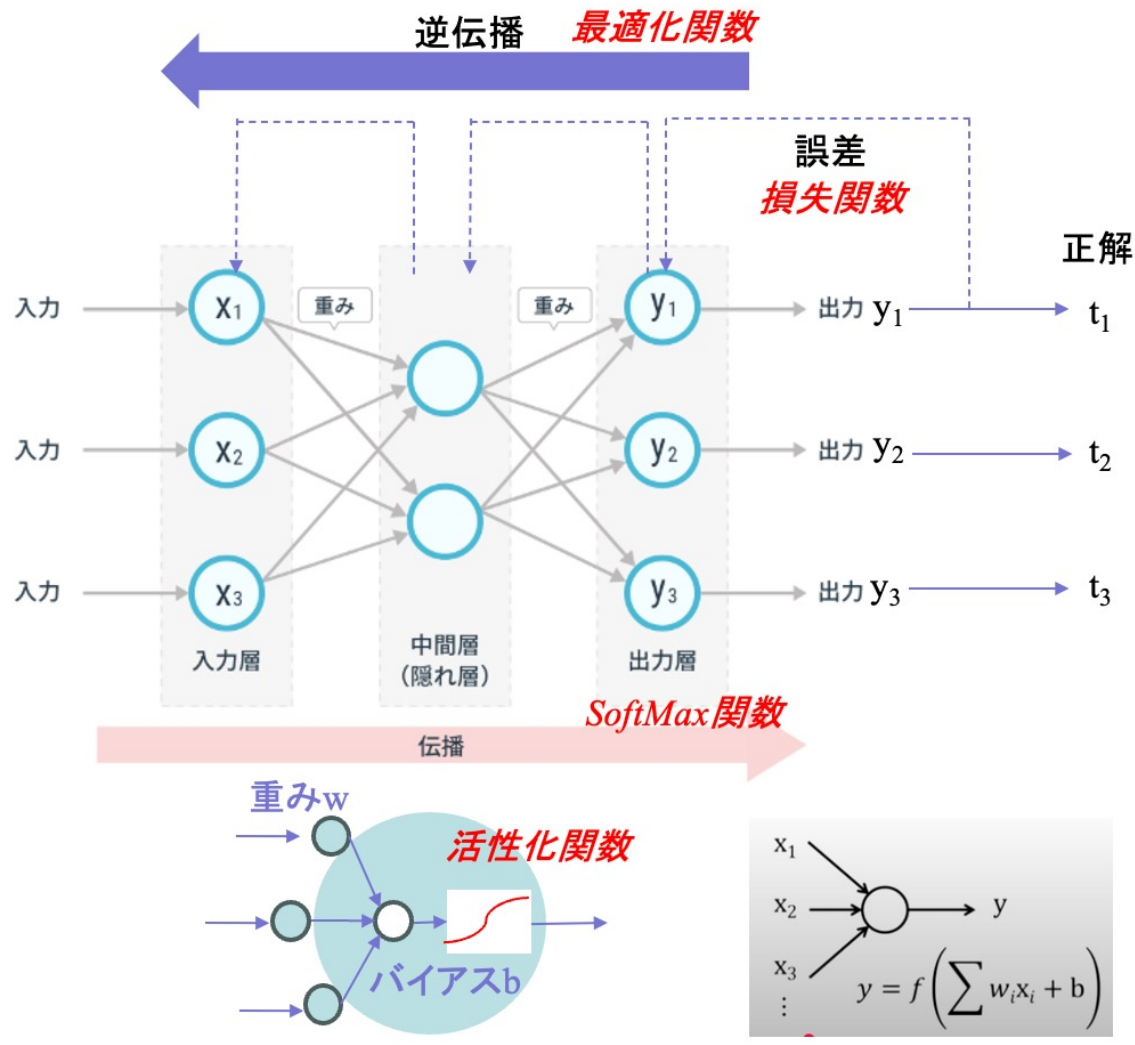
1.2. Neural Networkの学習とは？

計算対象:パラメータ

- 重み W
- バイアス b

固定値:ハイパーパラメータ

- 層数
- ニューロン数
- 活性化関数
- SoftMax関数
- 損失関数
- 最適化関数
- Epoch数
- Batch数



Neural Networkにおける
Parameter

Hyper Parameter (赤色)

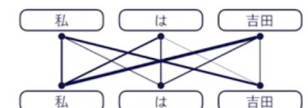
のイメージ図。

赤字の箇所は何の数学的関数を適用するのか決める必要がある。これがHyper Parameterで、実際にはPytorchのメソッドを選択することが作業。

Transformerの場合は、逆伝播は存在せず、Parameterは主にAttention行列になる。

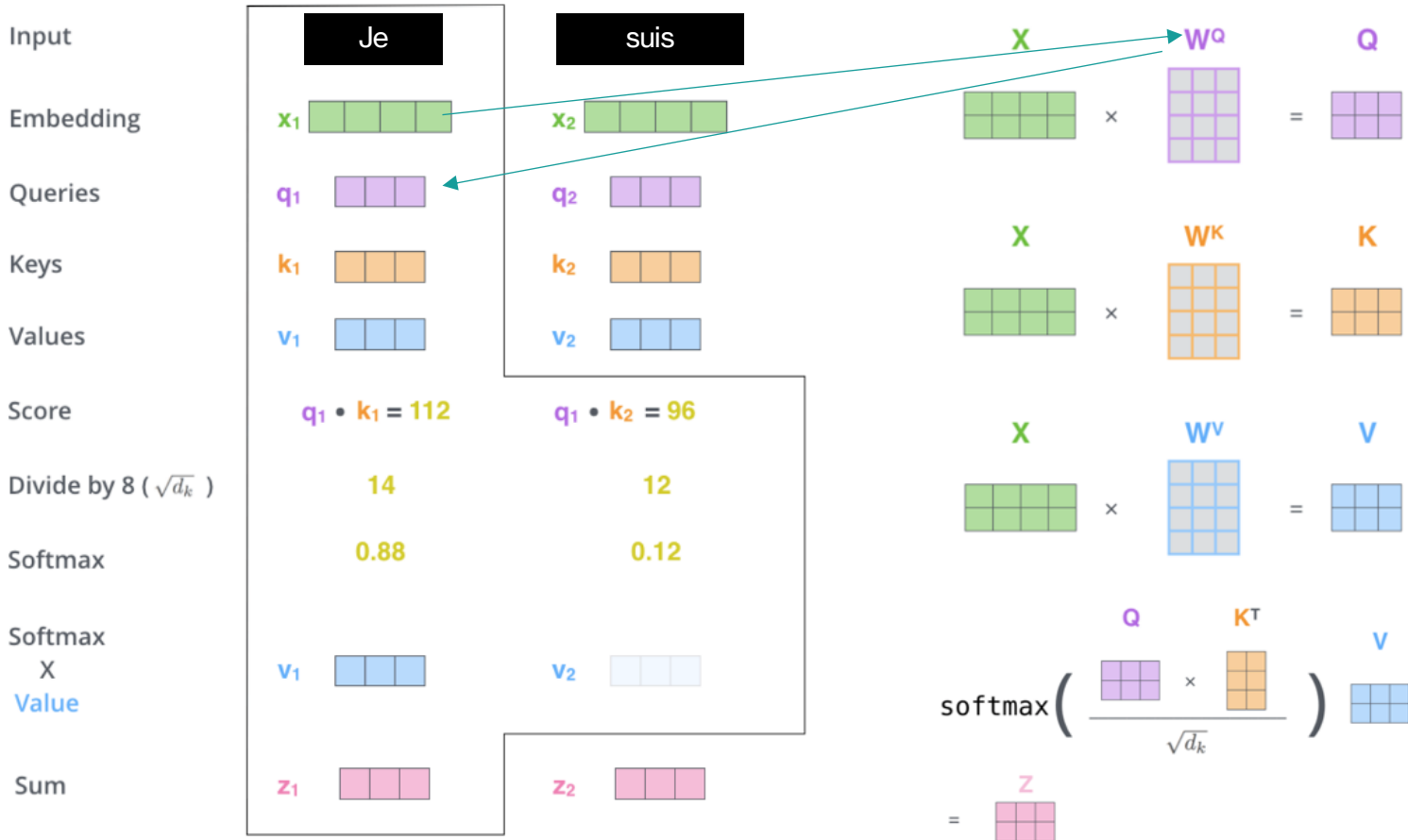
自己注意機構
self-attention mechanism

自系列間での重み付け



1.3. Transformerの学習とは？

Q, Vを持ちいてAttention Scoreを算出、Softmaxで確立に変換します。最後に、その確立とVを掛けて、Self-Attentionの値Zを求めます。Zはベクトルですが、最終的には文章全体的を表現するために行列Zになります。左は単語レベル、右は文章レベルでの計算のイメージ。



LLMのParameterである

Attentionとは何か？

入力単語間の相関を調べるために、
入力単語ベクトルを少し回転させる
係数の役目を果たす、

R単語ベクトル次元数 x 隠れ層の次元数

の行列 W^Q , W^K , W^V 。

この行列がLLMの学習Parameter。

このWがAttention。

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

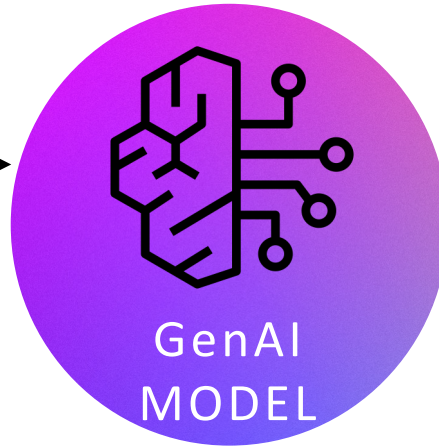
このW(Attention)を求めるための手段を指定しているのがLLMのHyper Parameter。

進化① 会話能力を獲得

”AWS上にHPを作りたいので
方法を教えて”



API Call



” AI Response

- VPCを作って
- Subnetを作って
- EC2をつかって...

AIが人間と対等に会話
ができる！

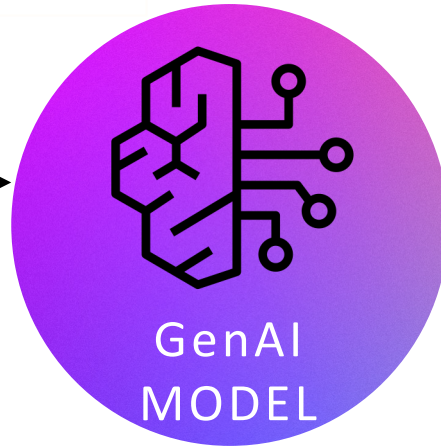
進化② AIに仕事をさせることが出来るようになる

”AWS上にHPを作りたいので
- VPCを作って
- Subnetを作って
- EC2をつくって...”



API Call

AI Response



”CFnのコード例です”

新しい情報検索ツール
としてLLMが使われる！

長いプロンプト書くのが面倒！

進化③ 思考能力を獲得 CoT (Chain of Thought)の発明

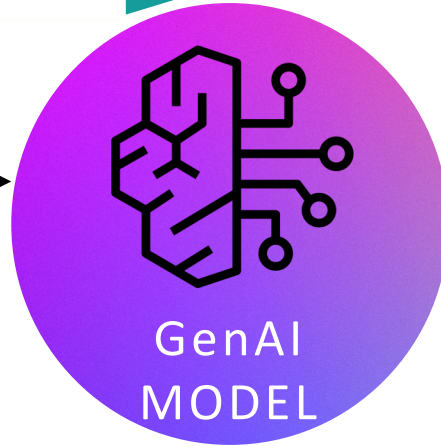
”AWS上にHPを作りたいので
- VPCを作って
- Subnetを作って
- EC2をつくって...”



API Call

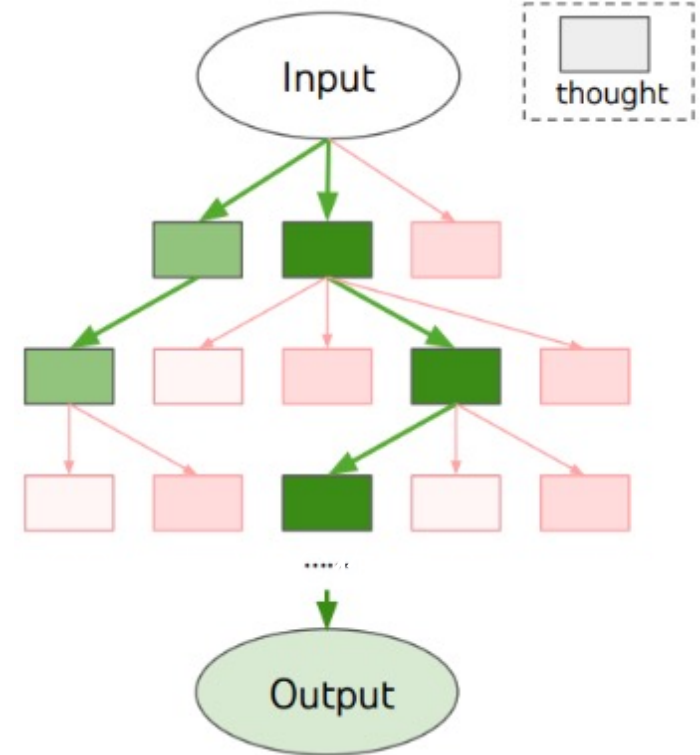


AI Response



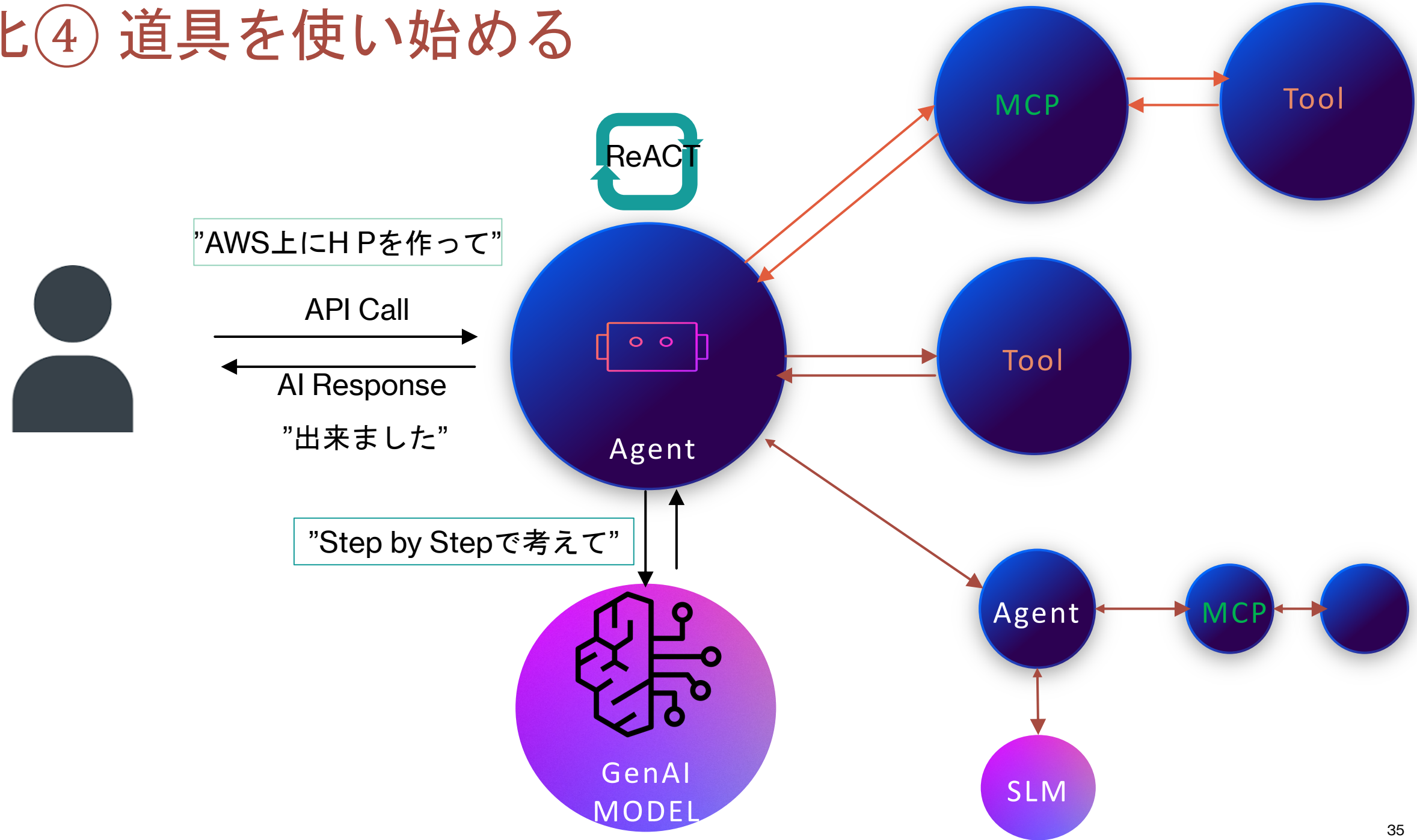
”CFnのコード例です”

”AWSでHPを作る方法をStep by Stepで考えて”

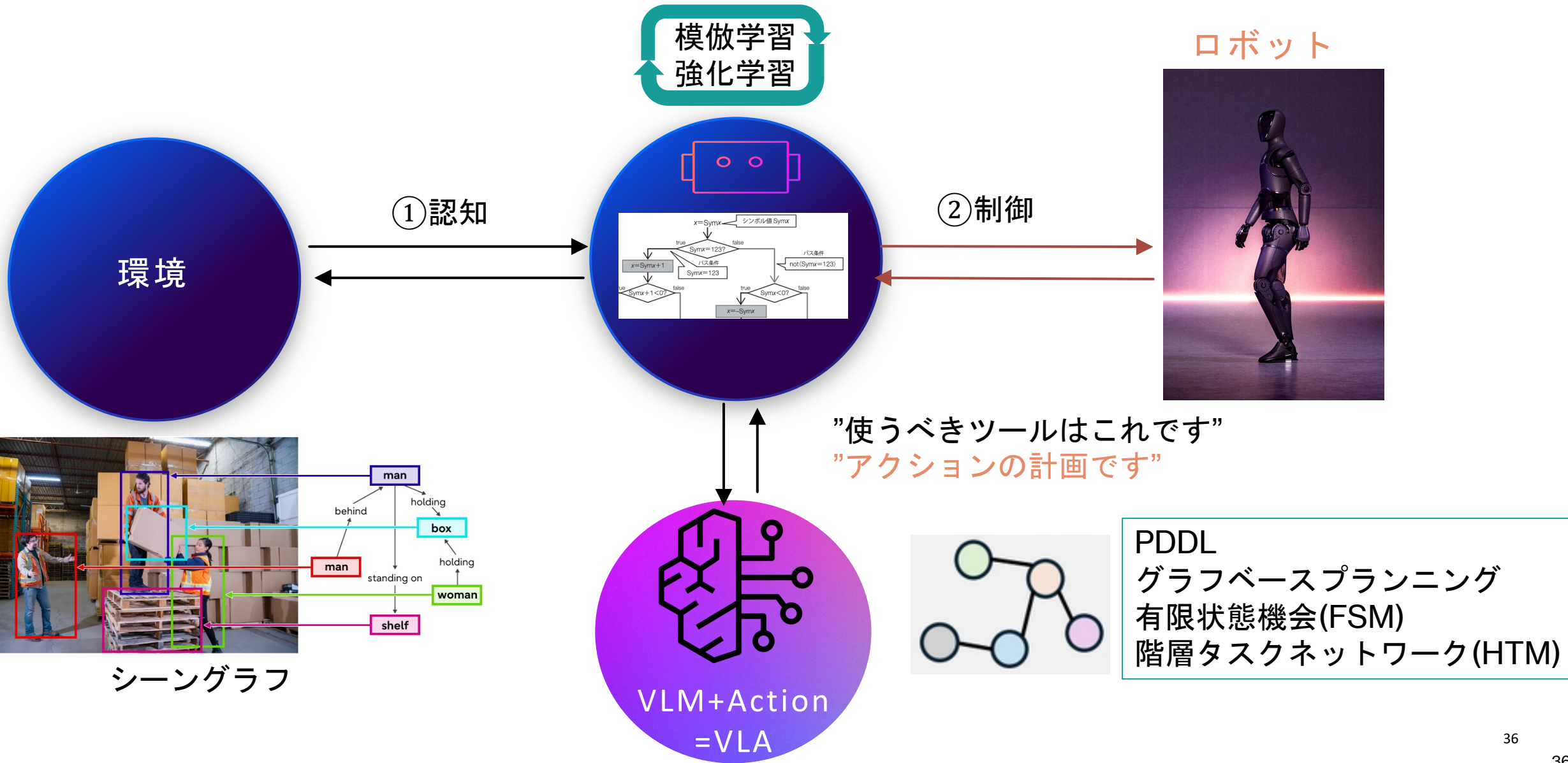


CoT: <https://arxiv.org/abs/2201.11903>
ToT: <https://arxiv.org/abs/2305.20050>
GoT: <https://arxiv.org/abs/2308.09687>
Process Supervision: <https://arxiv.org/pdf/2305.20050>

進化④ 道具を使い始める

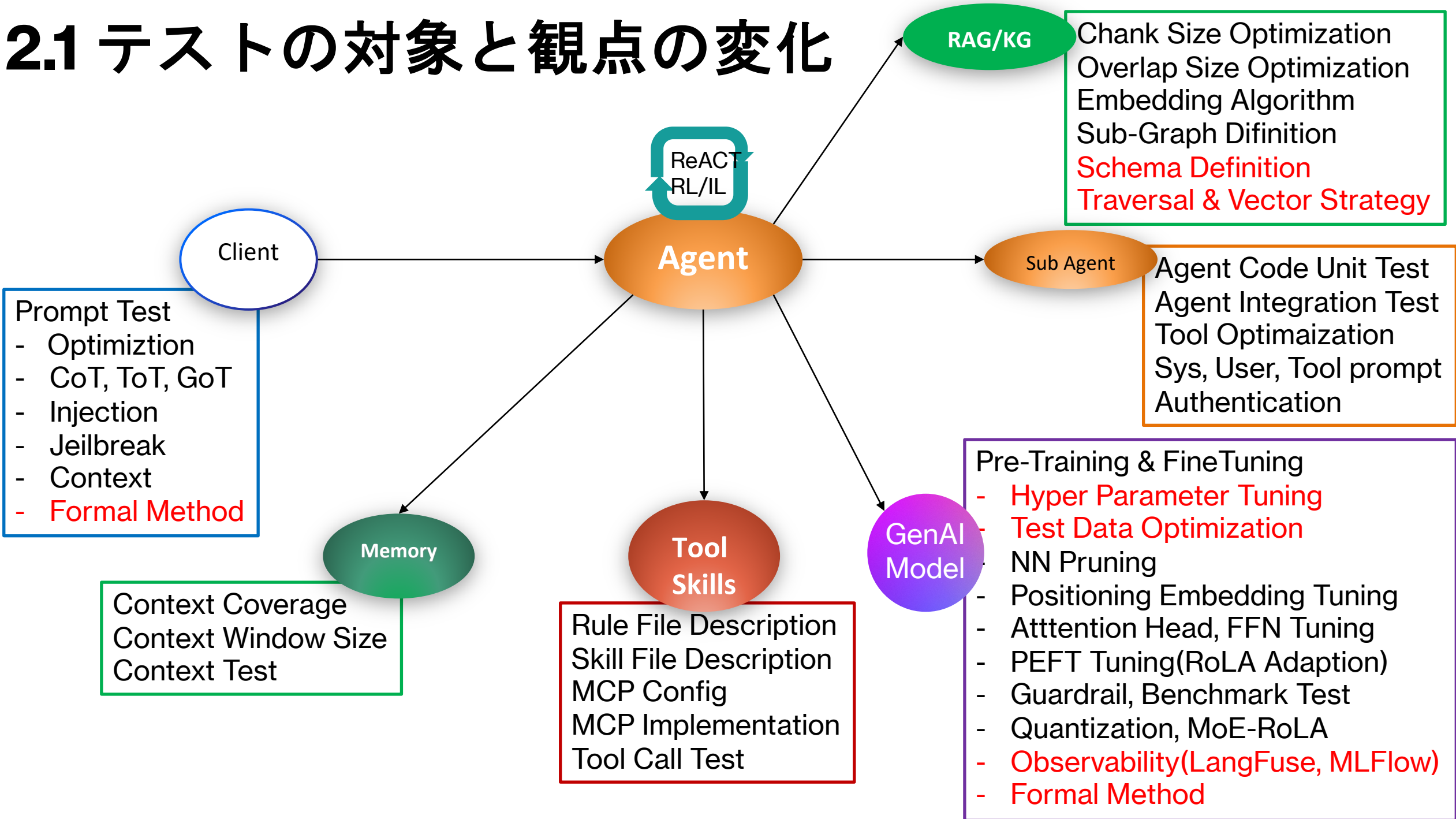


進化⑤体を手に入れる



2. 生成AI時代における テストエンジニアリング

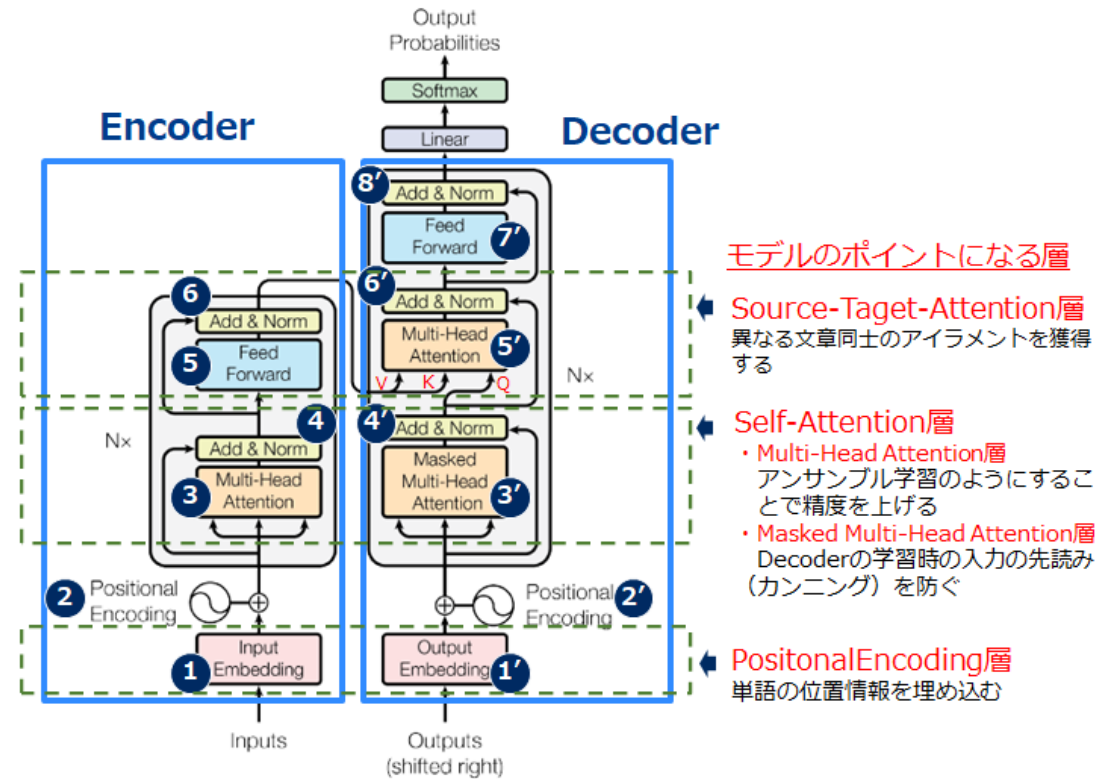
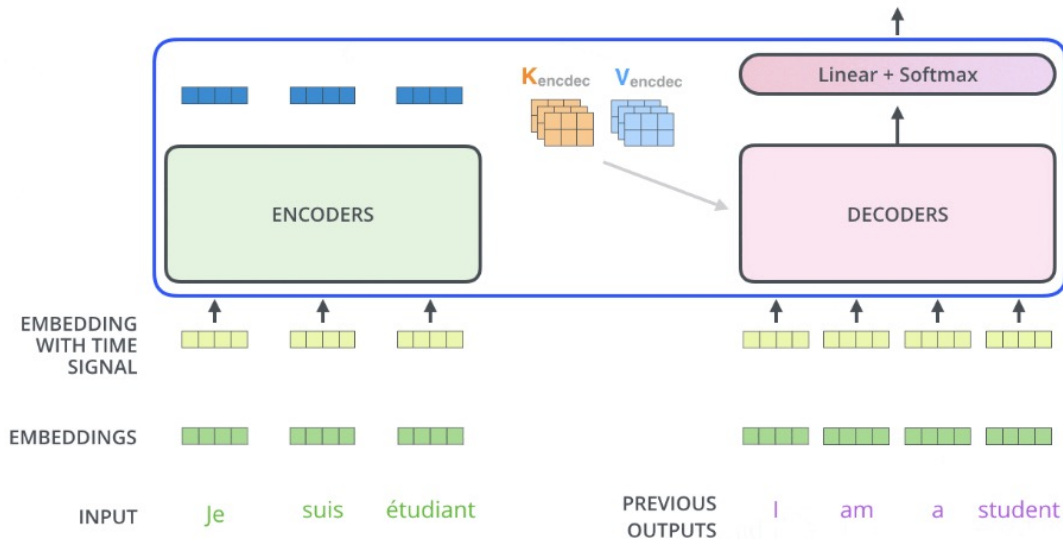
2.1 テストの対象と観点の変化



2.2 Transformer

生成AI時代においては、テストエンジニアはテスト対象である、Transformerアーキテクチャの実装、Hyper-Parameter, AI-Agent Frameworkの実装を理解する必要がある。

Decoding time step: 1 2 3 4 5 6 OUTPUT | am a student <end of sentence>



Appnedix

生成AIのソフトウェア開発およびテストへの利用例

研究動向の時系列整理（2022-2025）

- 2022年から2025年にかけて、生成AIのソフトウェア開発・テストへの応用研究が急速に進展。
- 2022年：初期応用・可能性検証フェーズ（ChatGPT試行レポートなど）
- 2023年：コード生成や品質・信頼性の課題整理に関する体系的な研究（IEEE Softwareなどで総説）
- 2024年：テスト設計・自動化・DevOps統合など実務応用プロトコル確立。
- 2025年：安全性・透明性・責任分界などを含む本格的な特集論文・レビュー発表、セキュリティやXAI(説明可能AI)の研究が拡大。

生成AIのソフトウェア開発およびテストへの利用例

- ソフトウェアライフサイクルにおいて、生成AIはコード生成、テストケース生成、デバッグ支援など幅広く応用されています。
- 代表的な学術論文・研究例：
 - Unifying the Perspectives of NLP and Software Engineering: A Survey on Language Models for Code (Zhang2023), コード生成・バグ検出・テスト自動化を体系的に整理
 - CodePori: Large-Scale System for Autonomous Software Development Using Multi-Agent Technology (Rasheed2024), 要求仕様から動くコードを生成する大規模自動化ワークフロー
 - From Defects to Demands: A Unified, Iterative, and Heuristically Guided LLM-Based Framework for Automated Software Repair and Requirement Realization (Liu2024), 自動バグ修復と要求実装、テスト駆動型LLM活用のフレームワーク

生成AIのソフトウェア開発およびテストへの利用例 - Webサイト・技術記事上の情報 -

- GitHub Articles: AI in Software Development
 - 設計・コード生成・テスト・運用などSDLC全体でのAI活用
- “知られざる”コード生成AI10選（2025年版）
 - DevinやCopilotなどのIDE統合型、テスト支援機能付きツールを紹介
- 生成AIが変革するソフトウェア品質保証の未来（日本総研）
 - QA工程での自動テストケース生成や高速なバグ検出支援例を報告
- AIがコード生成の一部を担う実例（Microsoft / Google / Meta）
 - 実際に30%以上のコードをLLMによって生成する事例が注目

生成AIのソフトウェア開発およびテストへの利用例

- 企業導入事例 -

- 各企業で、生成AIを活用して開発速度やテスト効率の向上が報告されています。
- 事例：
 - 富士通：GitHub Copilot導入により、リファクタリングで作業量約75%削減。
 - NTTデータ：テスト効率化により生産性3倍。CI/CD環境と自動テスト連携。
 - 大規模Eコマース企業（非公開）：回帰テストサイクルを80%短縮、デイリーデプロイが実現。
 - 銀行系企業：自動化されたコンプライアンステストにより監査準備時間が数週間→数時間に。

ご清聴ありがとうございます。

ソフトウェアテスト技術振興協会
(ASTER)調査研究事業の活動の
紹介でした。

ご清聴ありがとうございます。