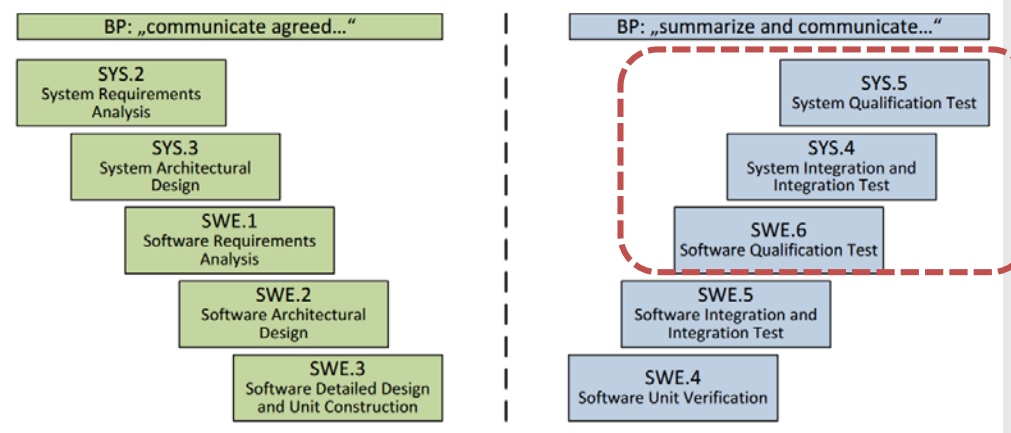
JaSST'16 Tokyoテクノロジーセッション

AUTOSAR Acceptance Testの自動化の取り組み

株式会社ベリサーブ
オートモーティブ検証サービス開発部
須原秀敏

自己紹介

項目	内容
所属	株式会社ベリサーブ
名前	須原秀敏（すはらひでとし）
経歴	車載電子機器 （以後ECU）の システムテストを6年
活動	WACATE、STAC、JaSST、SQiP、TEF東海etc...に参加、講演経験はなし
SNS	Twitter:@suhahide



http://www.automotivespice.com/fileadmin/software-download/Automotive_SPICE_PAM_30.pdfより

アジェンダ

1. AUTOSARとは？
2. AUTOSAR Acceptance Test（AT）とは？
3. ATの自動化
4. 考察と今後の課題

アジェンダ

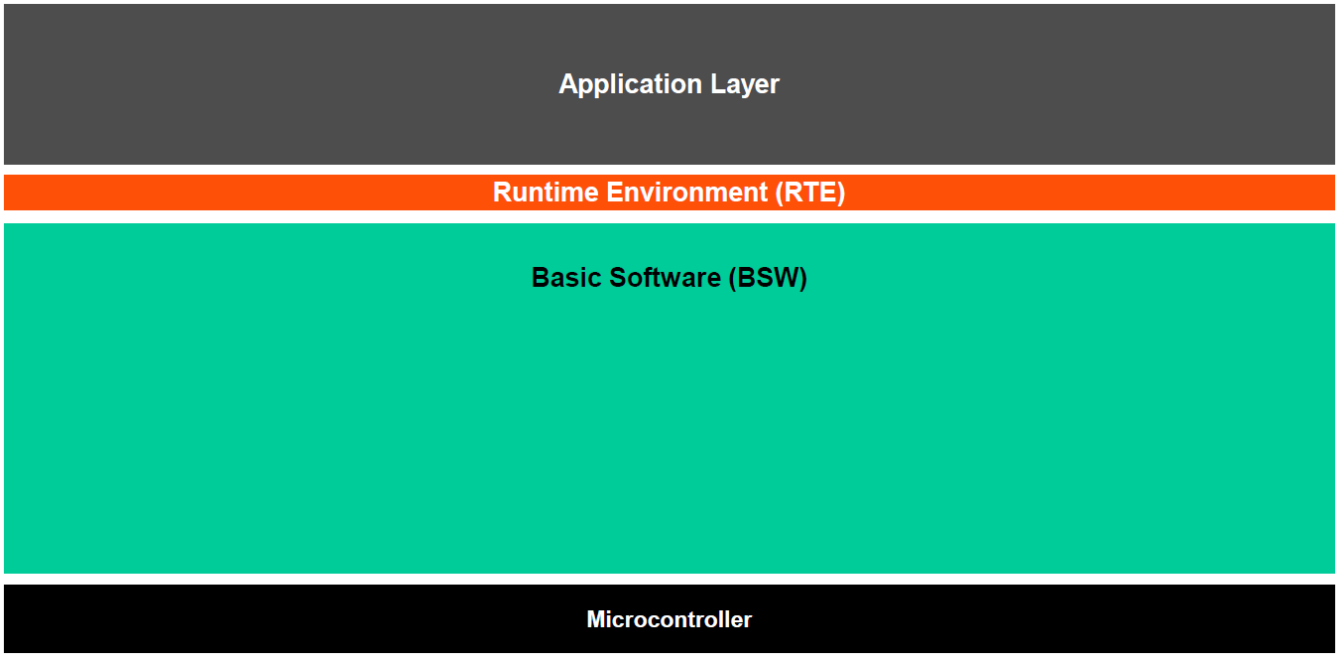
1. AUTOSARとは？
2. AUTOSAR Acceptance Test（AT）とは？
3. ATの自動化
4. 考察と今後の課題

AUTOSARとは

項目	説明
正式名称	AUTomotive Open System ARchitecture
略称	AUTOSAR
いつ	2003年発足、現在Release4.2.2
誰が	OEMやサプライヤ、ツールベンダなどのグローバルパートナーシップ
対象	ECU
どの部分	ソフトウエアアーキテクチャ
なぜ	ECUの開発コストを下げ、品質を確保
どうする	非競争領域は共通化、競争領域は再利用と再配置

AUTOSARのソフトウエアアーキテクチャ(1)

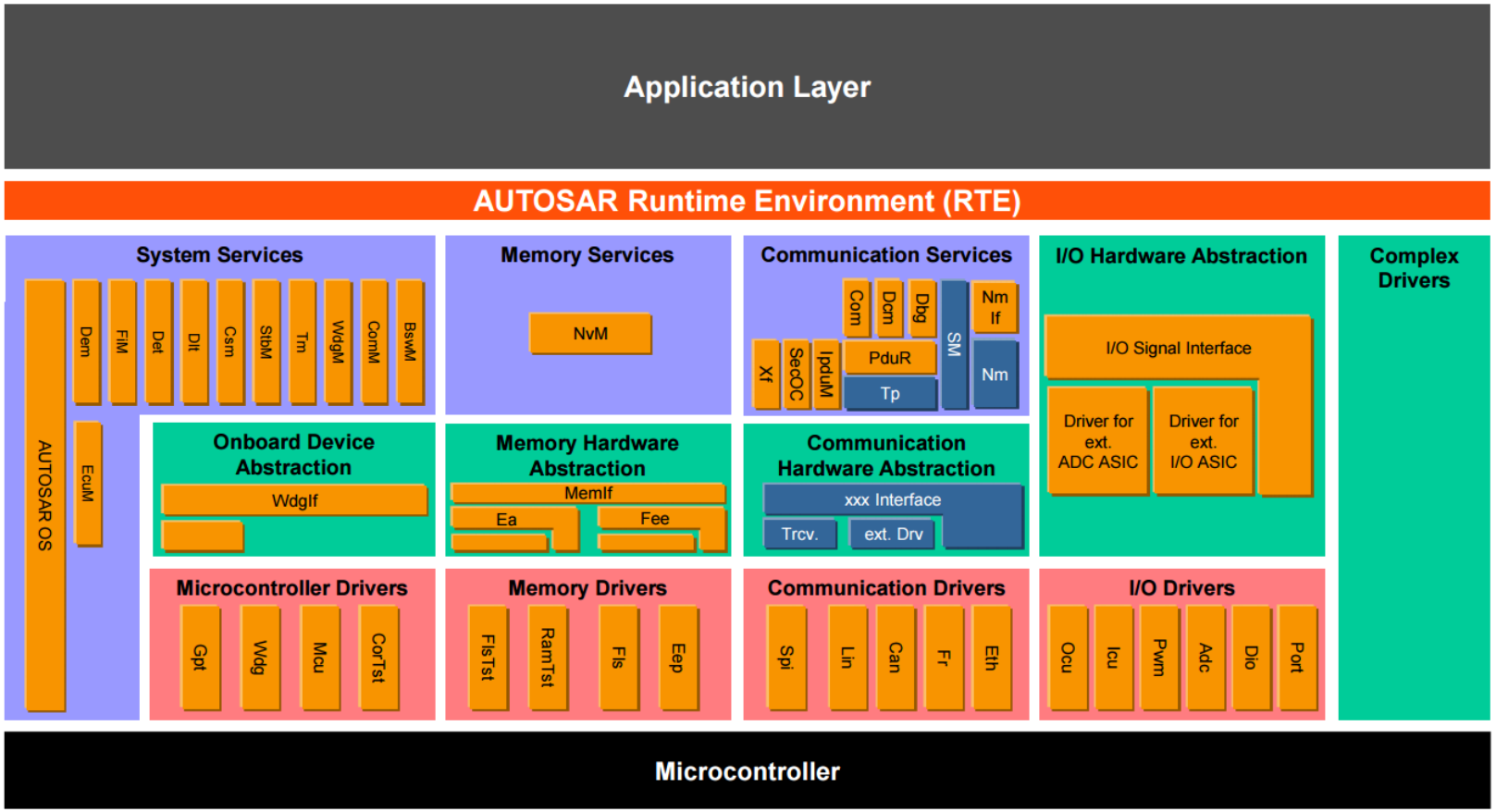
Layer	略称	説明
Application Layer	APP	ECUの機能を実現。一つ以上のSWCから構成される
Runtime Environment	RTE	Application Layerへの実行環境提供、通信IFの提供
Basic Software	BSW	どんなECUにも共通である土台部分の基本ソフトウエア



http://www.autosar.org/fileadmin/files/releases/4-2/software-architecture/general/auxiliary/AUTOSAR_EXP_LayeredSoftwareArchitecture.pdfより

AUTOSARのソフトウエアアーキテクチャ(2)

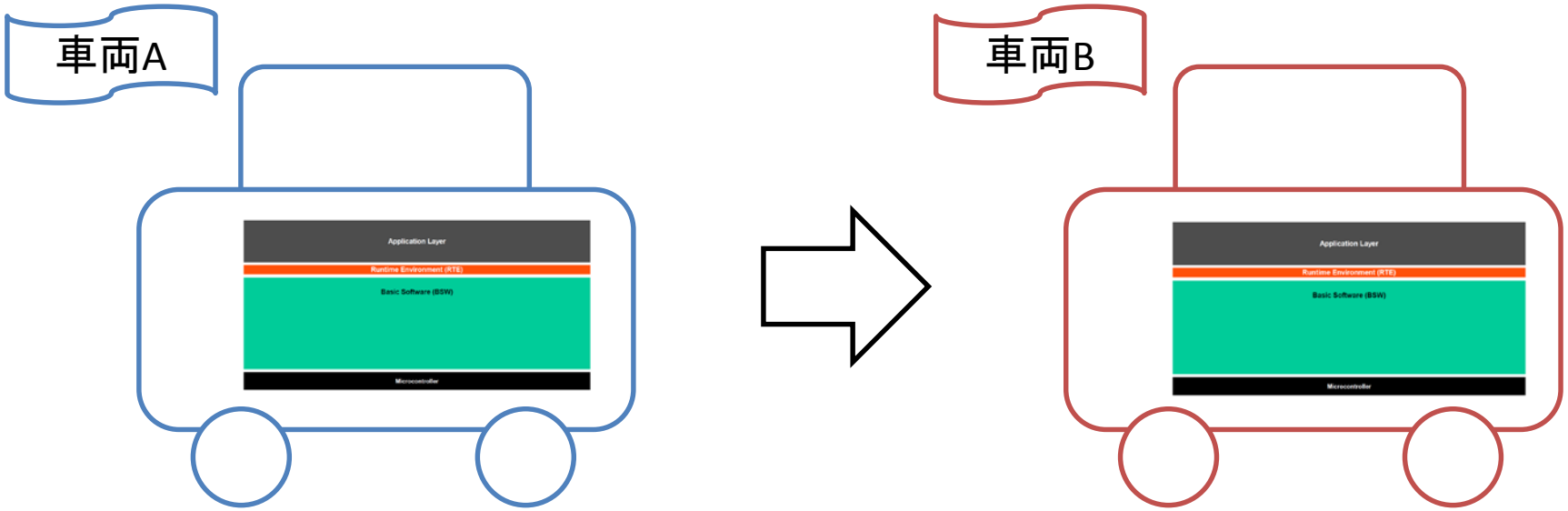
BSWの構成モジュール



http://www.autosar.org/fileadmin/files/releases/4-2/software-architecture/general/auxiliary/AUTOSAR_EXP_LayeredSoftwareArchitecture.pdfより

AUTOSARのメリット(1)

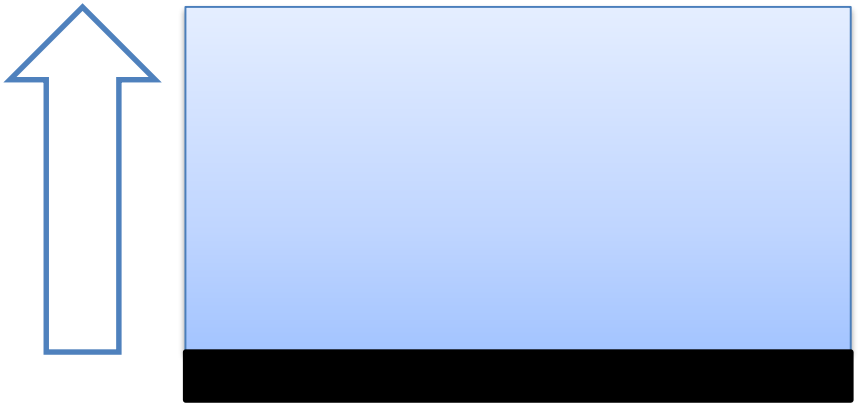
No	メリット	理由
1	車両の特性を越えて、Application Layerの再利用ができる	RTEによりAPI抽象化がなされる、また、外部からの設定により特性が吸収できる



AUTOSARのメリット(2)

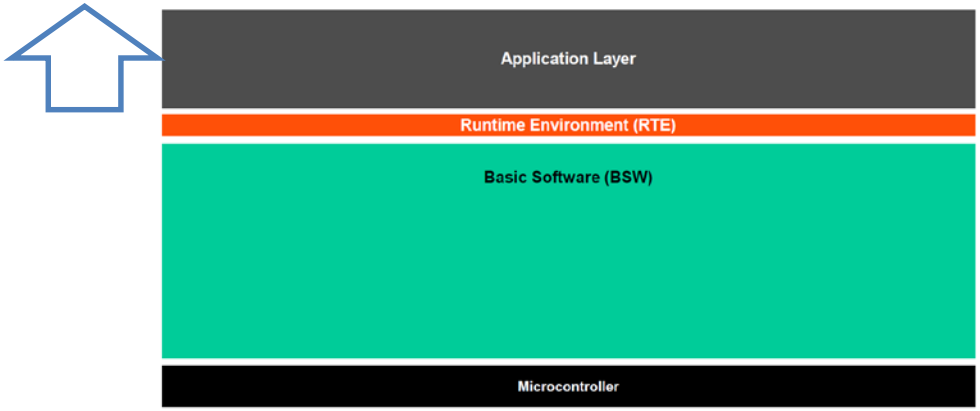
No	メリット	理由
2	安定した品質のプラットフォームを使用できる	BSWにはすべてのECUに共通のモジュールが含まれている

AUTOSARでない



マイコン部分除きゼロから開発

AUTOSAR

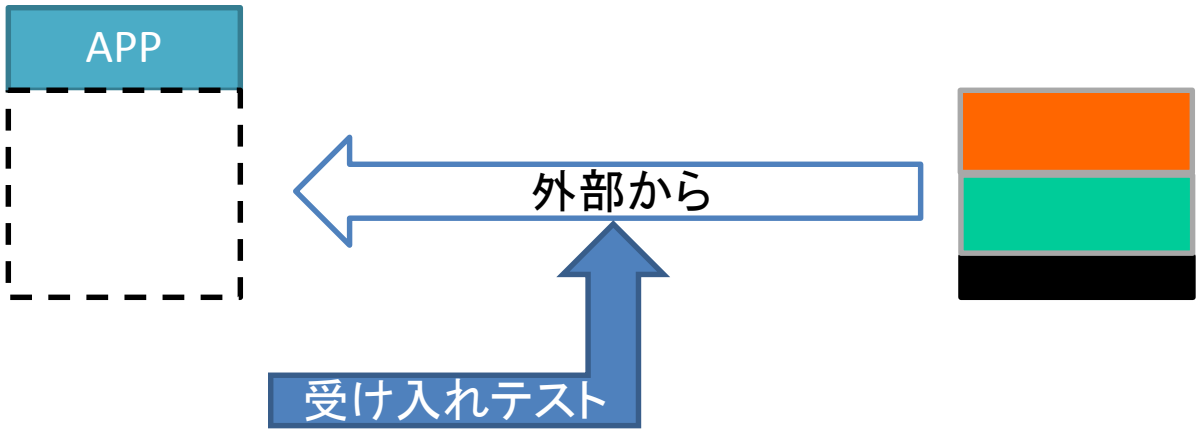


Application Layer＋設定での開発

AUTOSAR対応の製品のテストのニーズ

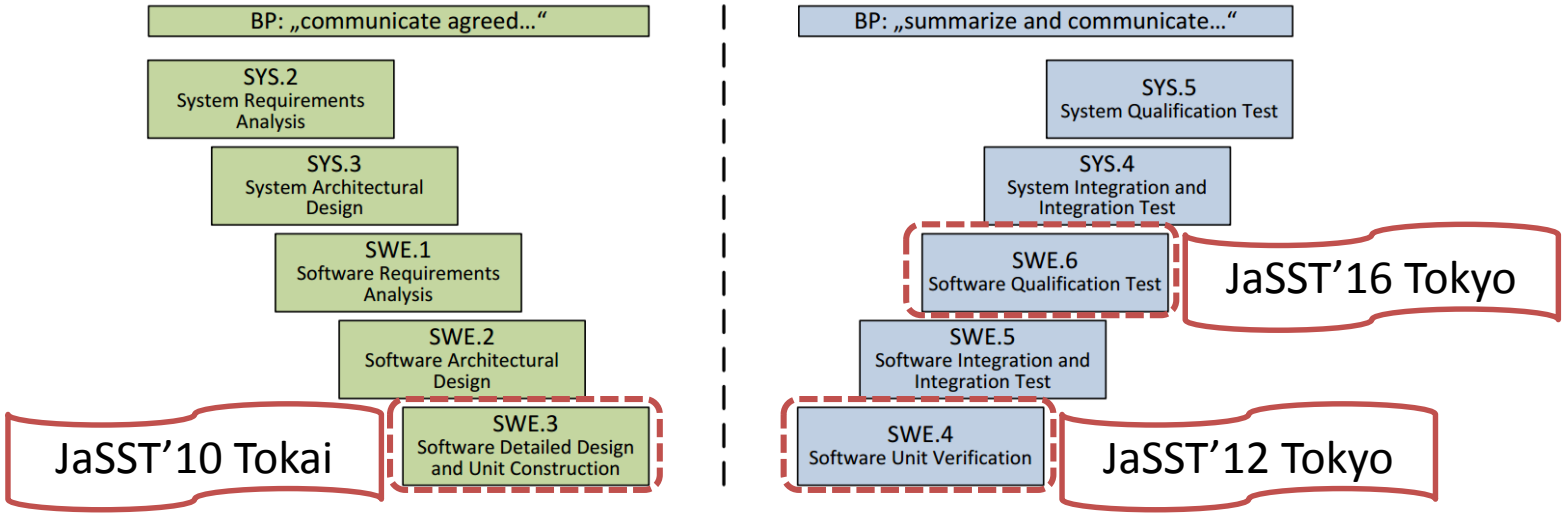
AUTOSARのコンセプト上、AUTOSAR対応BSW＋RTEを外から持ってきて、製品に組み込むことが多い

この場合、持ってきた製品の受け入れテストを実施したいというニーズが発生する



ちなみに：AUTOSARとJaSST

いつ、どこで	題名	概要
JaSST'10 Tokai	「AUTOSARを適用した車両システム開発環境の構築」	開発プロセスとツールチェーンの整備
JaSST'12 Tokyo	「AUTOSAR_OSに対するテストケースおよびテストプログラムの自動生成」	PictMasterによる単体テスト設計、実装の自動化
JaSST'16 Tokyo	「AUTOSAR Acceptance Testの自動化の取り組み」	システムテスト実行の自動化



アジェンダ

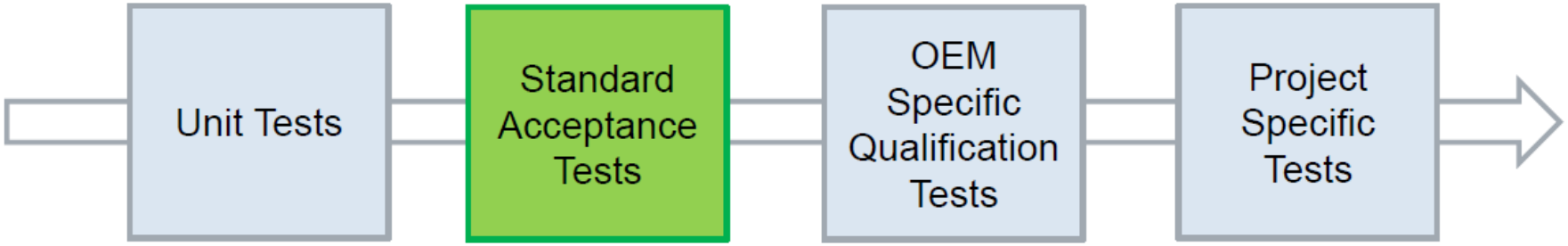
1. AUTOSARとは？
2. AUTOSAR Acceptance Test（AT）とは？
3. ATの自動化
4. 考察と今後の課題

AUTOSARのAcceptance Testとは？

項目	説明
正式名称	AUTOSAR Acceptance Test
略称	AT
いつ	2014年リリース、現在Release1.1
誰が	AUTOSAR
対象	AUTOSARの提供する「機能」
どの部分	受け入れテスト
なぜ	テスト実施の苦労とコストを削減
どうする	共通の受け入れテストを定義

ATの目的と制限

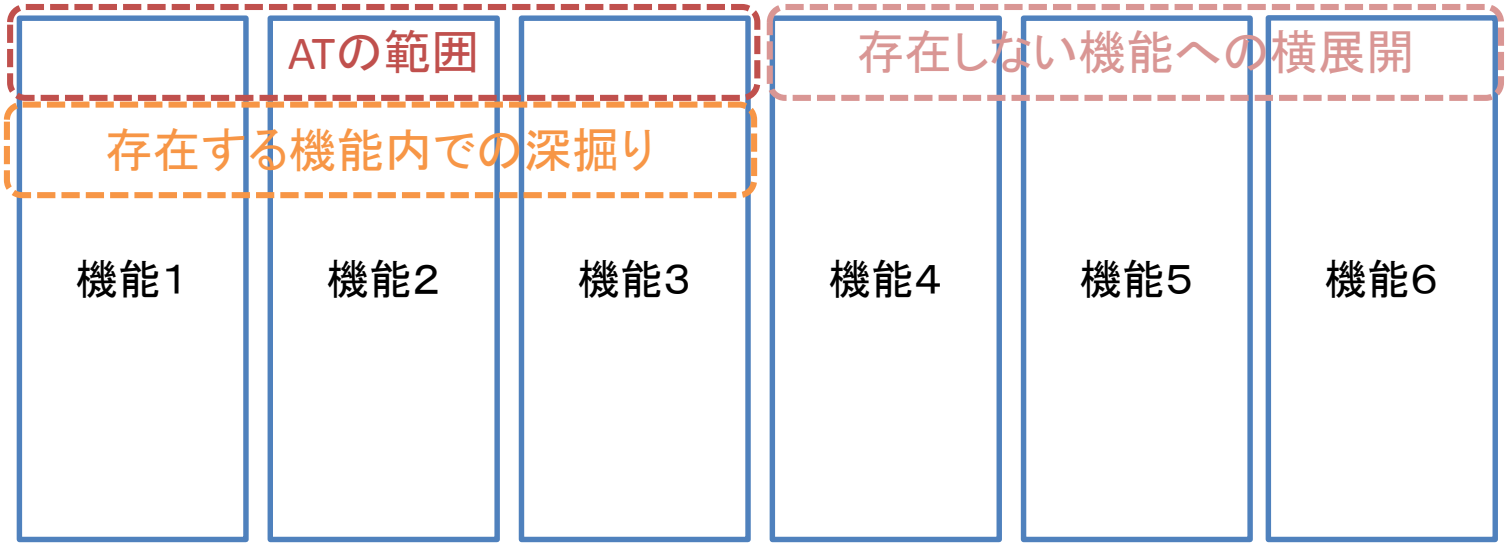
分類	内容	説明
目的	共通化	共通のテスト開発とメンテナンス。個別にテストを作る必要がない
	方向付け	メソドロジー、拡張性を提供。カバレッジ外のテストを作る際に方向性が決まる
	結果の流用	テスト実施結果の流用。サプライヤとOEM両方でテスト実施する必要がない
制限	カバレッジ	限定された機能のみを含めている
	対応要否	OPTIONAL
	製品固有	プロジェクト固有の設定についてはテストしない



http://www.autosar.org/fileadmin/files/releases/tc-1-1/AUTOSAR_EXP_AcceptanceTestsOverview.pdfより

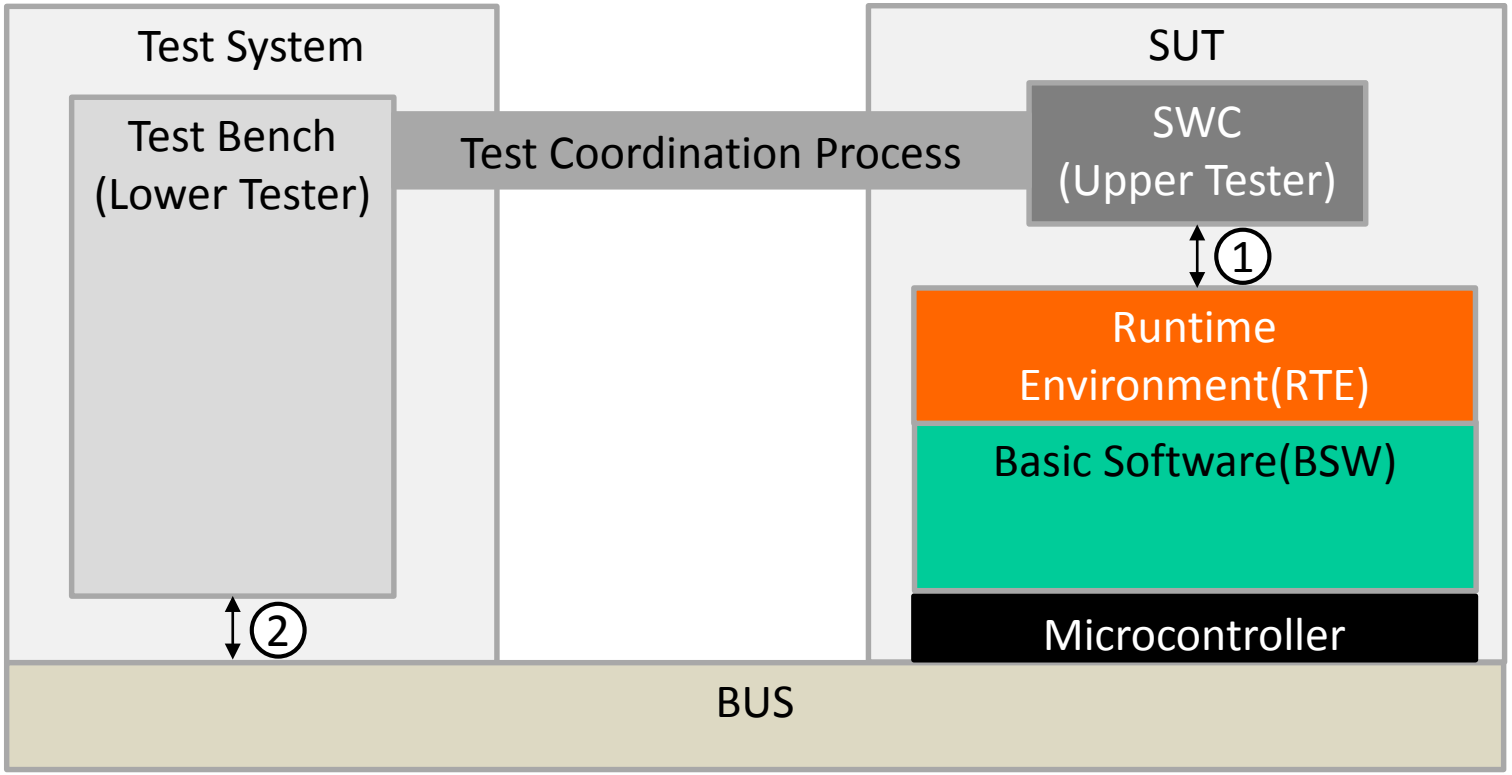
ATの使い方の例

シチュエーション例	使い方
受け入れテスト実施	ATで定義されているテストケースに加え、AT内に存在する機能内での深掘り、AT内に存在しない機能への横展開
回帰テスト	ATで定義されているテストケース＋αを毎回リリース時、毎回ビルド時に実施する



ATのテストアーキテクチャ

番号	説明
①	Upper Testerと呼ばれ、テスト対象のAPI部分のIF
②	Lower Testerと呼ばれ、テスト対象のバス部分のIF。CAN、LIN、DIO、ADCなどが含まれる



ATのテストケースのサンプル

Test Steps		Pass Criteria
Step 1	***SWC:*** Request ModeSwitch (call to BswMModeRequest port) to IPDU_ACTIVATED (start ComIPduGroup AT_208_IpduGroup)	***BUS:*** AT_208_Ipdu is sent out after Offset Time. Next AT_208_Ipdu are sent out every Period Time AT_208_Sg1 value is AT_208_Sg1_Value_Init AT_208_Sg1 update bit is 0
Step 2	***SWC:*** Update signal AT_208_Sg1 (call Rte_Write() API for Port AT_208_Sg1) with AT_208_Sg1_Value_1	***BUS:*** AT_208_Ipdu Periodic Time is not changed (value is Period Time) AT_208_Sg1 UpdateBit is set to 1 in the first send, after that it is 0 AT_208_Sg1 value is now AT_208_Sg1_Value_1
Step 3	***SWC:*** Invalidate signal AT_208_Sg1 (by calling API Rte_Invalidate())	***BUS:*** AT_208_Ipdu Periodic Time is not changed (value is Period Time) AT_208_Sg1 UpdateBit is set to 1 in the first send, after that it is 0 AT_208_Sg1 value is now AT_208_Sg1_Value_Invalid
Post-conditions	Not applicable	

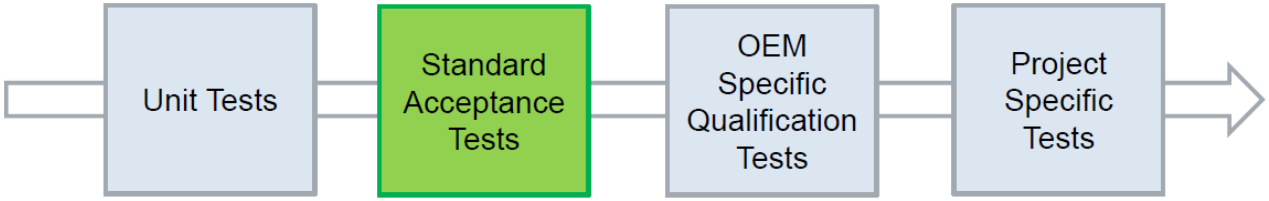
テスト手順が記載されている。本事例ではAPIコールのみ

期待結果が記載されている。本事例ではBUS出力の確認のみ

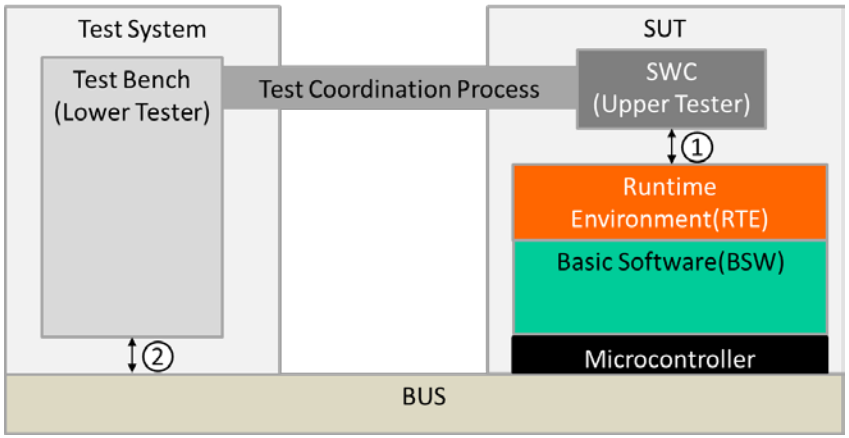
http://www.autosar.org/fileadmin/files/releases/tc-1-1/specifications_auxiliary/AUTOSAR_ATS_CommunicationCan.pdfより

ATの特性まとめ

No	特性
1	基本的な機能確認のテスト
2	手順が明確に定義されている
3	再実施されやすい



Test Steps		Pass Criteria
Step 1	*SWC:* Request ModeSwitch (call to BswMModeRequest port) to IPDU_ACTIVATED (start ComIPduGroup AT_208_IpduGroup)	*BUS:* AT_208_Ipdu is sent out after Offset Time. Next AT_208_Ipdu are sent out every Period Time AT_208_Sg1 value is AT_208_Sg1_Value_Init AT_208_Sg1 update bit is 0
Step 2	*SWC:* Update signal AT_208_Sg1 (call Rte_Write() API for Port AT_208_Sg1) with AT_208_Sg1_Value_1	*BUS:* AT_208_Ipdu Periodic Time is not changed (value is Period Time) AT_208_Sg1 UpdateBit is set to 1 in the first send, after that it is 0 AT_208_Sg1 value is now AT_208_Sg1_Value_1
Step 3	*SWC:* Invalidate signal AT_208_Sg1 (by calling API Rte_Invalidate())	*BUS:* AT_208_Ipdu Periodic Time is not changed (value is Period Time) AT_208_Sg1 UpdateBit is set to 1 in the first send, after that it is 0 AT_208_Sg1 value is now AT_208_Sg1_Value_Invalid
Post-conditions	Not applicable	

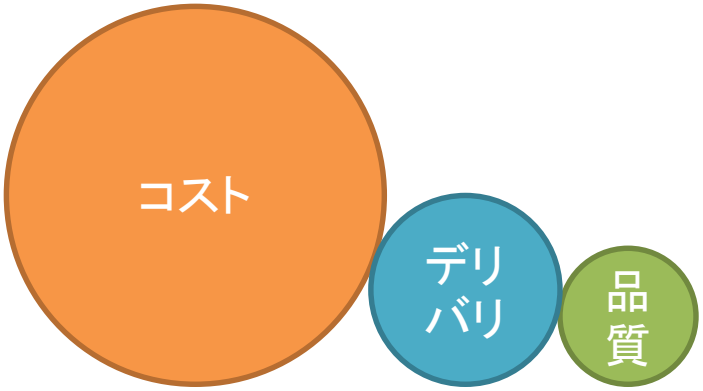


アジェンダ

1. AUTOSARとは？
2. AUTOSAR Acceptance Test（AT）とは？
3. ATの自動化
4. 考察と今後の課題

一般的な自動テストのメリットと今回のゴール

対象	説明
品質	テスト実行品質の向上。手動ではできないデータ量やタイミングなど
コスト	再実施のコスト削減、デグレード削減によるコスト削減
デリバリ	開発ライフサイクルの加速



今回はコスト削減をゴールに自動化の導入を考えた

ATと自動テストの相性

ゴール：コスト削減したい

そのためには…

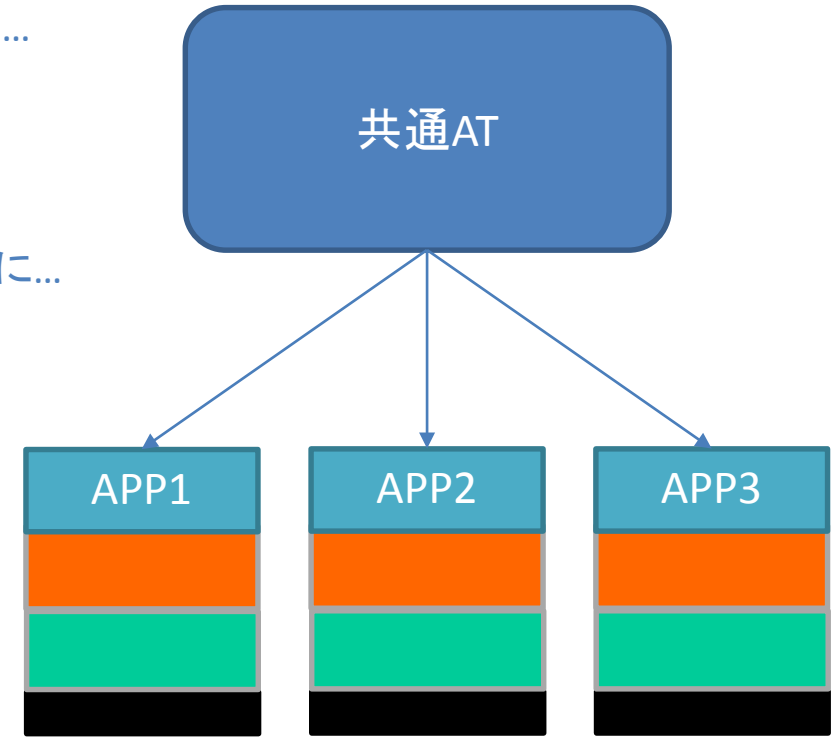
同じテストを複数回実施する

ATで考えると…

ATは共通部分に対するテストであるため、複数のECUに対してひとつのテストで実施できる

ゆえに…

ATの特性と自動テストの目的が一致



実現系：ATの自動テストの実装

項目	説明
実装戦略	後述
Test Bench側ツール	National Instruments社のVeriStand+LabVIEW
対象AT	Release1.1全件（200件強）

Content of Acceptance Tests Release R1.0.0

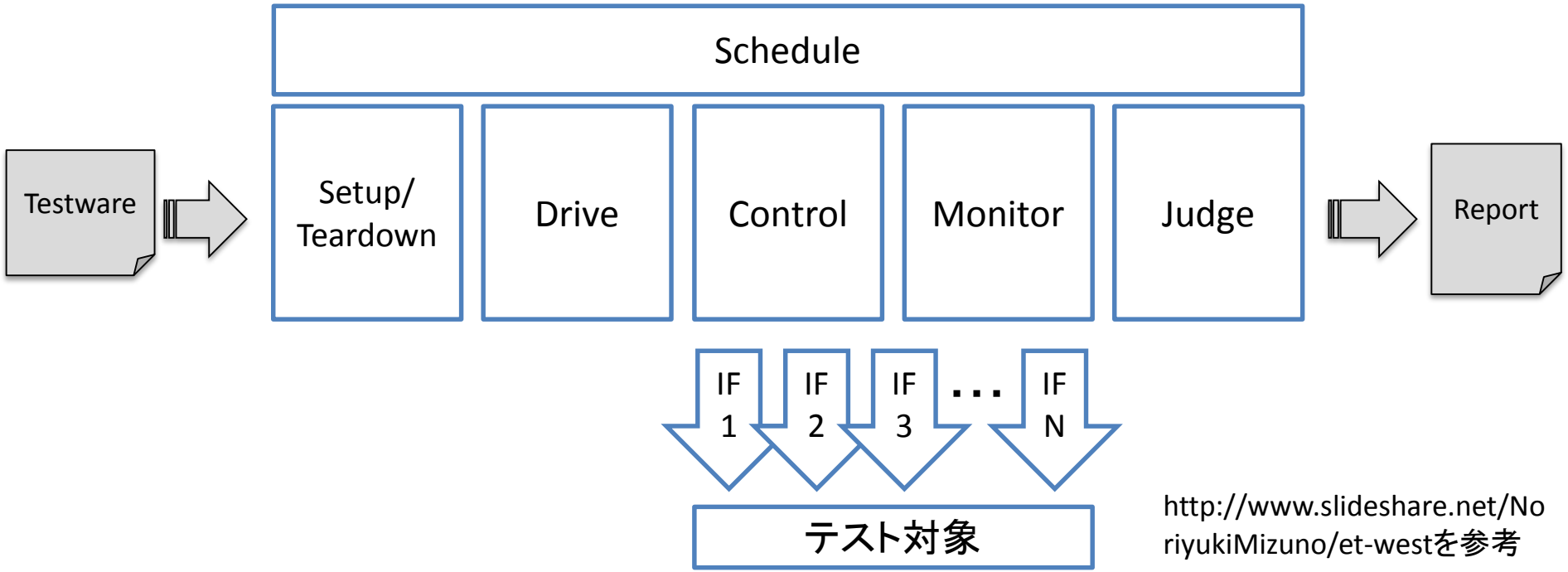
Document	# TC	Features	Short Description
AUTOSAR_ATS_CommunicationCan	6	RS_BRF_01592	Data Transfer
	2	RS_BRF_01648	Large Data Type
	6	RS_BRF_01707	Can Bus Off handling
AUTOSAR_ATS_CommunicationFlexRay	6	RS_BRF_01592	Data Transfer
	2	RS_BRF_01648	Large Data Type
AUTOSAR_ATS_CommunicationLin	6	RS_BRF_01592	Data Transfer
	2	RS_BRF_01648	Large Data Type
AUTOSAR_ATS_CommunicationManagement	5	RS_BRF_01448	ComM Current Mode
	10	RS_BRF_01680	Network management (Fr)
	4	RS_BRF_01680	Network management (Lin)
	10	RS_BRF_01688	ComM User Request
	4	RS_BRF_01696	Partial Networking
	14	RS_BRF_01680	Network management (Can)
AUTOSAR_ATS_CommunicationViaBus	6	RS_BRF_01600	Timeout Handling
	22	RS_BRF_01616	Initial Values
	4	RS_BRF_01632	Data Consistency
	13	RS_BRF_01592	Data Transfer (Bus independent)
	6	RS_BRF_01648	Large Data Type
AUTOSAR_ATS_DiagnosticServices	11	RS_BRF_02184	DiagnosticMonitor (DEM)
	8	RS_BRF_02144	DataServices (DCM)
	2	RS_BRF_02144	RoutineServices (DCM)
AUTOSAR_ATS_EcuModeManagement	2	RS_BRF_01488	EcuM Current Mode
	3	RS_BRF_01488	EcuM State Request
	2	RS_BRF_02152	EcuM Boot Target
	3	RS_BRF_02152	EcuM Shutdown Target
AUTOSAR_ATS_MemoryStack	15	RS_BRF_01416	NvM services
AUTOSAR_ATS_RTE	19	RS_BRF_01312	Rte Client Server Feature
		RS_BRF_01320	
	6	RS_BRF_01328	Rte SWC scheduling and activation from events
	20	RS_BRF_01376	Rte Data Conversion Feature
		RS_BRF_01304	
	13	RS_BRF_01352	Rte Sender Receiver Communication

http://www.autosar.org/fileadmin/files/releases/tc-1-1/AUTOSAR_EXP_AcceptanceTestsOverview.pdfより

一般論：自動テストアーキテクチャ

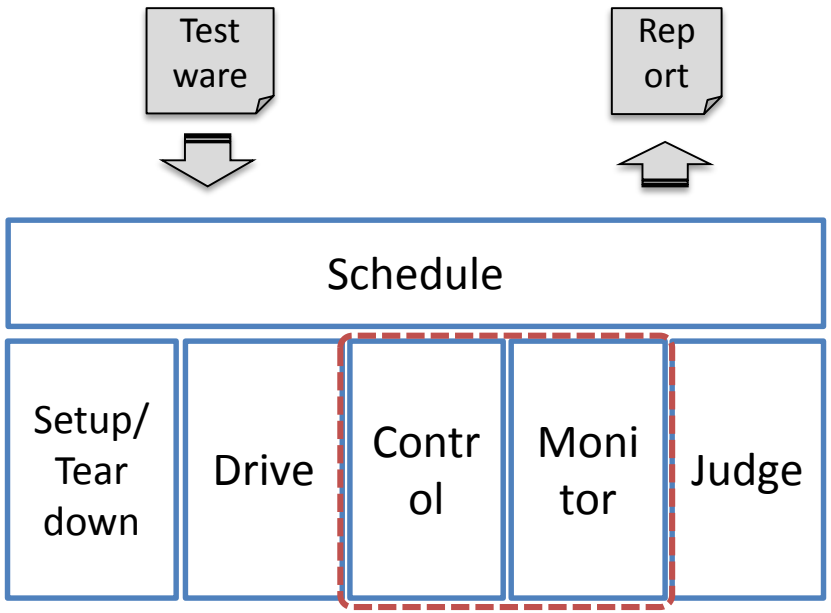
名称	説明
Schedule	テスト管理
Setup/ Teardown	前後処理
Drive	テスト手順進行

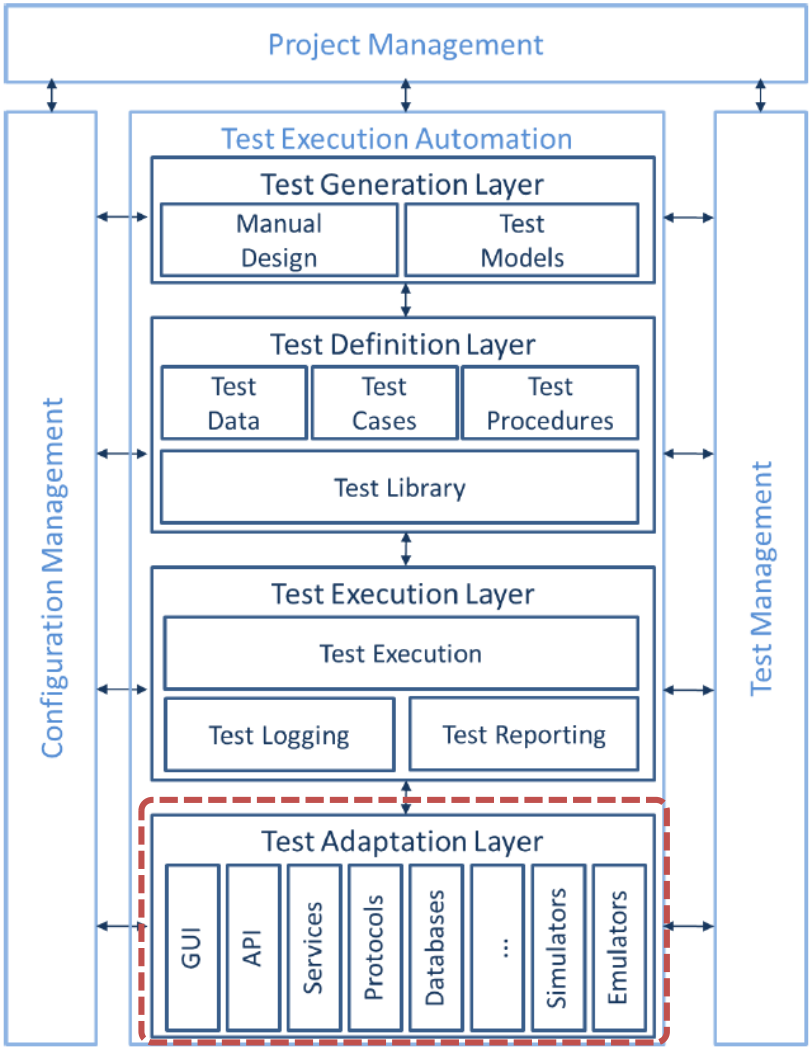
名称	説明
Control	テスト対象制御
Monitor	テスト対象からの入力監視
Judge	期待値との比較



http://www.slideshare.net/NoriyukiMizuno/et-westを参考

一般論：ISTQBのgTAAとの比較

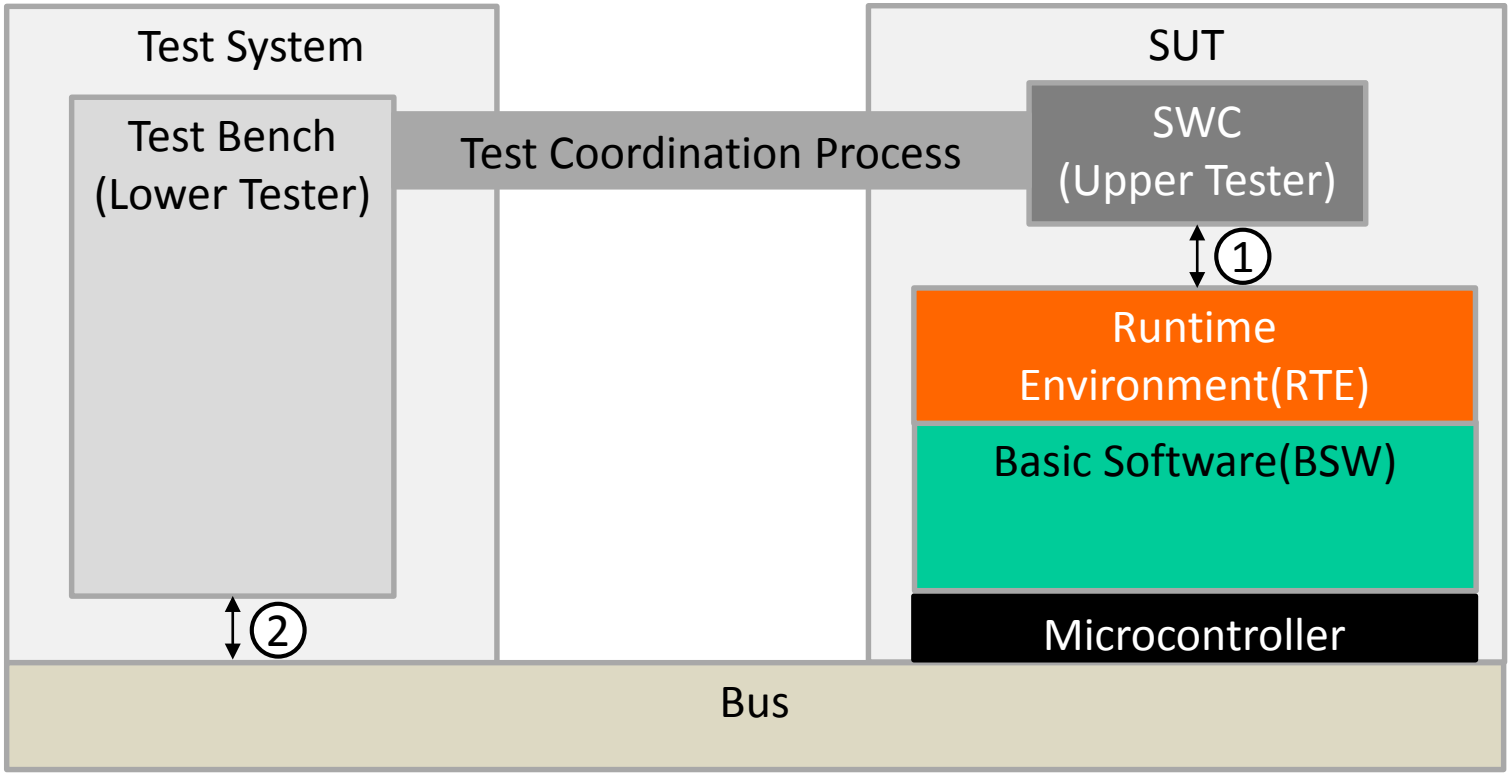




expert_level_syllabus_-_test_automation_-_engineering.pdfより

一般論：ATのテストアーキテクチャ（再掲）

番号	説明
①	Upper Testerと呼ばれ、テスト対象のAPI部分のIF
②	Lower Testerと呼ばれ、テスト対象のバス部分のIF。CAN、LIN、DIO、ADCなどが含まれる

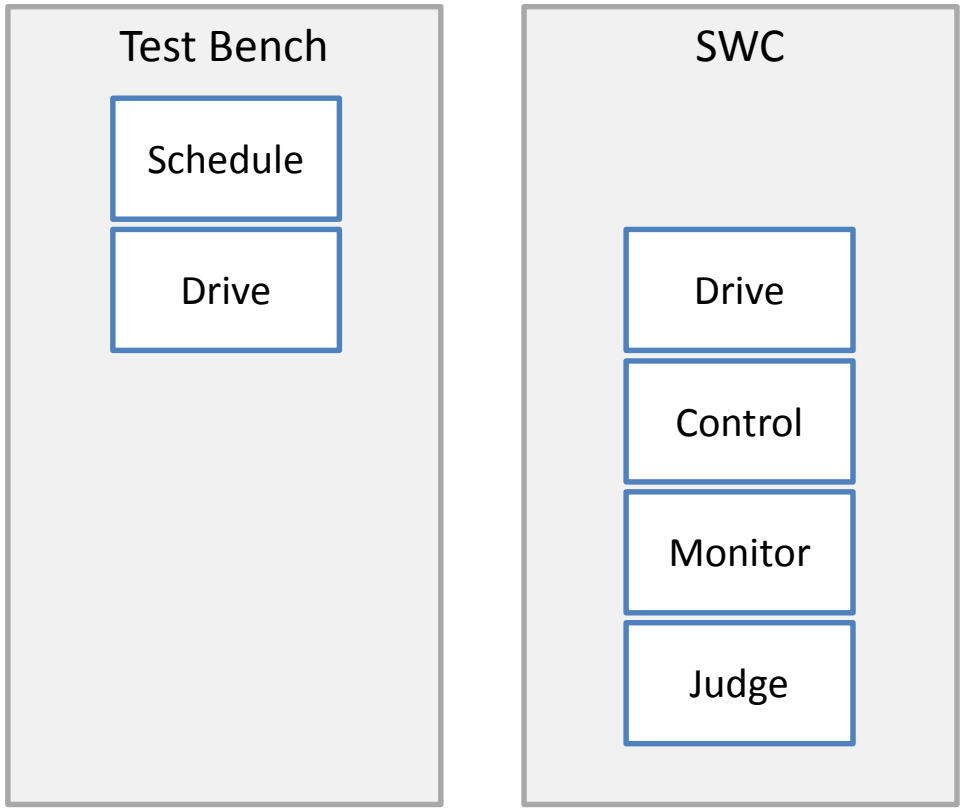


実現案：ATの自動テストアーキテクチャの評価観点

項目	説明
スクリプト数	TestBench側、SWC側合わせてどのぐらいのテストスクリプトを作成する必要があるか（今回は100件のテストケースを想定）
TCPの複雑さ	TCPにどの程度の量の情報を流す必要があるか
機能性	TestBench側が十分にScheduleを行うことができるための情報を得られるか、また、実施したいテストが実現できるか（タイミングなどの観点）

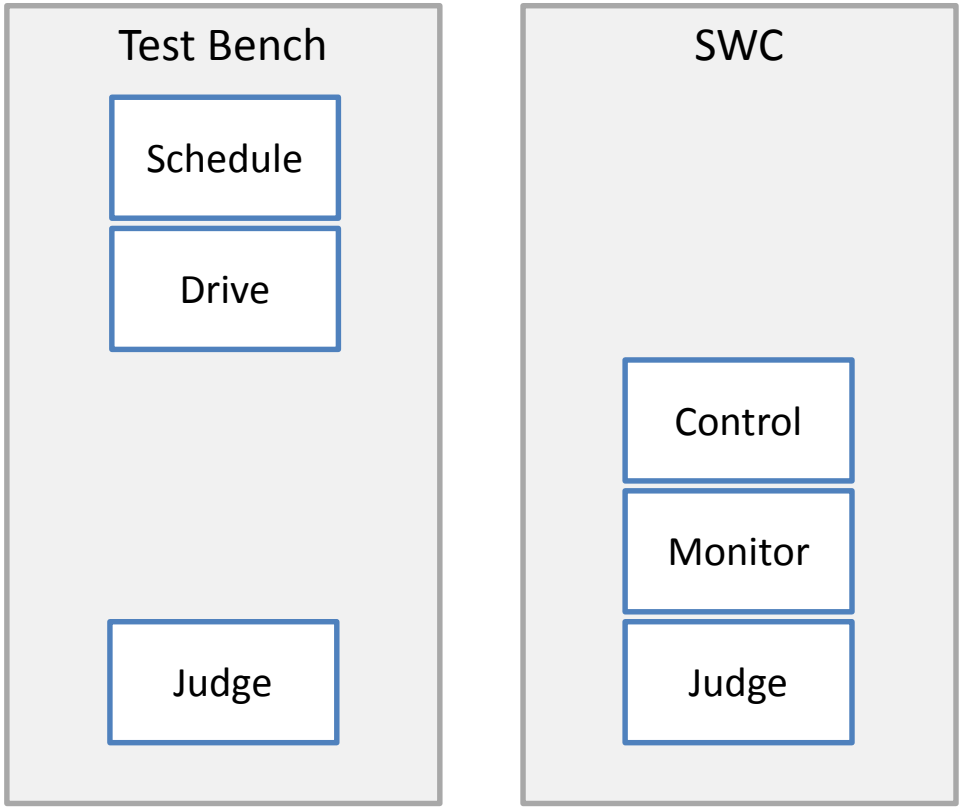
実現案：IF①（１）

項目	説明
コンセプト	全てSWC側に配置する
内容	最初のトリガのみTest Bench側から与え、その後のテスト手順は全てSWC側で実施する。Test Bench側は“Judge”の判定結果のみ受け取り、Reportする
スクリプト数	200件
TCPの複雑さ	単純
機能性	不足（ロギングに問題がある）



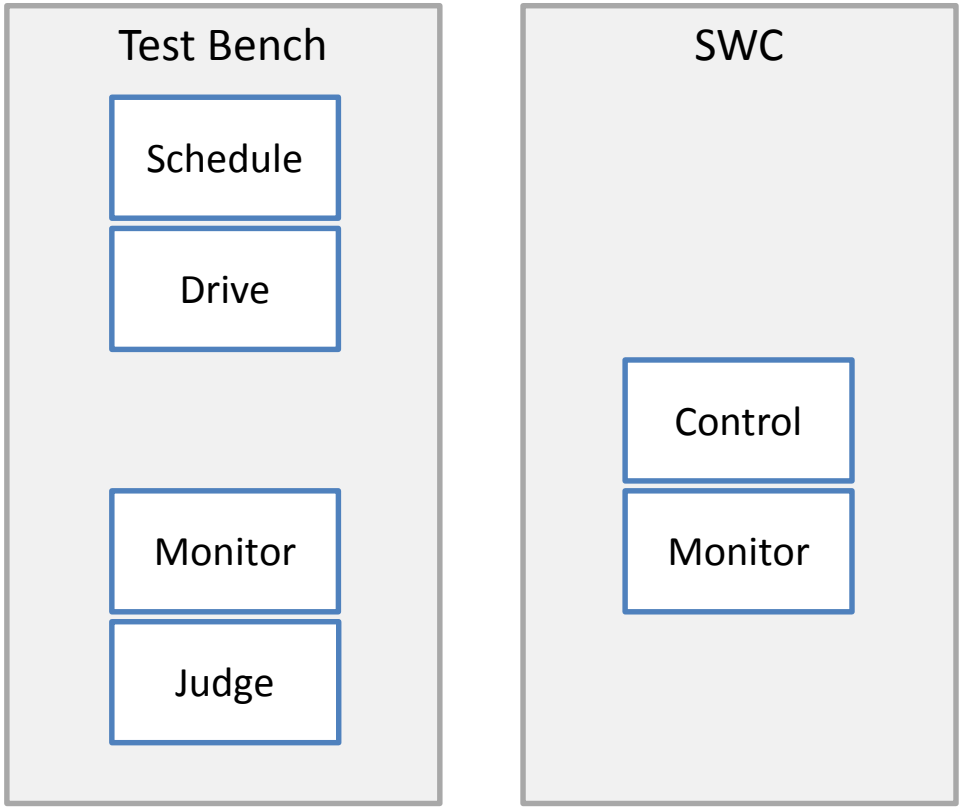
実現案：IF①（２）

項目	説明
コンセプト	簡易判定のみTest Benchに持たせる
内容	制御としてはトリガのみTest Bench側から与え、その後のテスト手順はSWC側で実施する。Test Bench側は“Judge”を一部実施する＋“Judge”の判定結果を受け取り、Reportする
スクリプト数	200件
TCPの複雑さ	単純
機能性	最低限



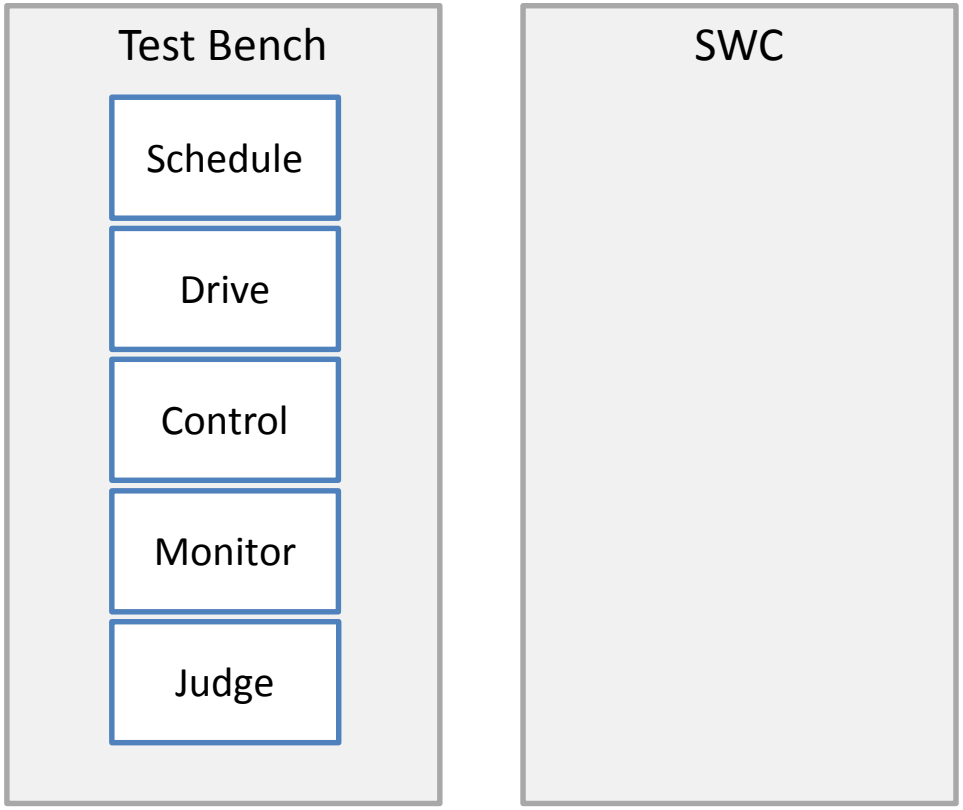
実現案：IF①（３）

項目	説明
コンセプト	全結果判定をTest Benchに持たせる
内容	制御としてはトリガのみTest Bench側から与え、その後のテスト手順はSWC側で実施する。Test Bench側は“Judge”をすべて実施し、Reportする
スクリプト数	200件
TCPの複雑さ	少し複雑
機能性	十分



実現案：IF①（４）

項目	説明
コンセプト	制御含めて全てTest Benchに持たせる
内容	SWC側は土管として存在し、全ての制御、判定をTest Bench側が持つ
スクリプト数	100件+1件
TCPの複雑さ	複雑
機能性	完全



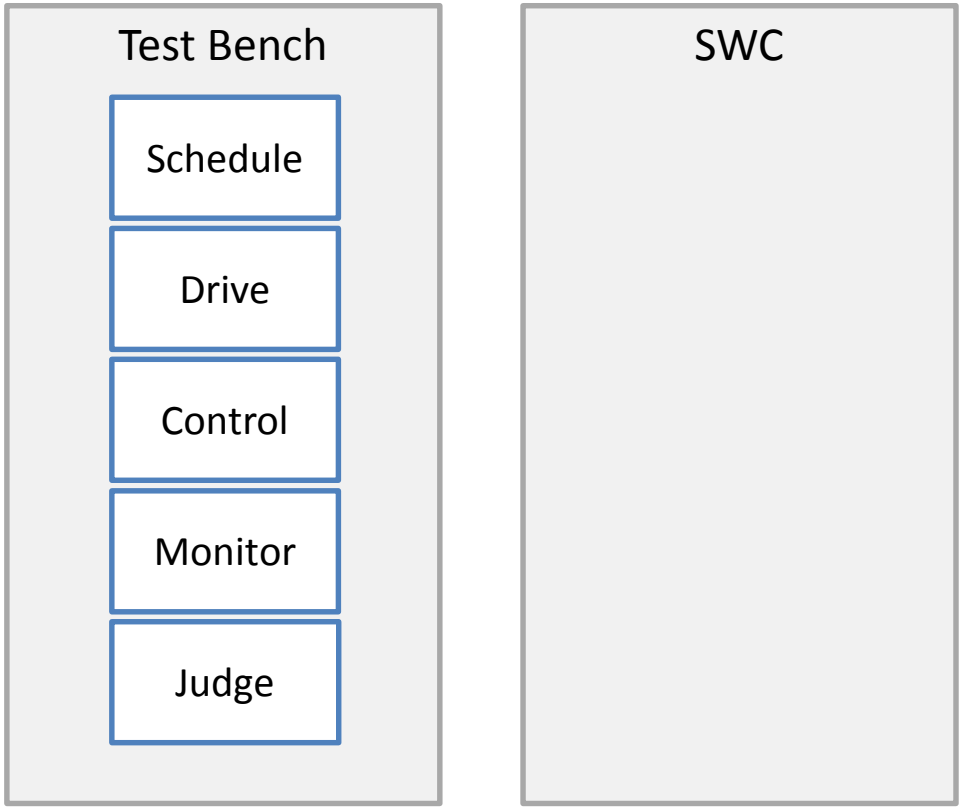
実現案：IF①実現案まとめ

番号	コンセプト	スクリプト数	TCPの複雑さ	機能性	点数
（1）	全てSWC	×	◎	×	3
（2）	簡易判定をTest Benchに	×	◎	△	4
（3）	全判定をTest Benchに	×	△	○	3
（4）	全てTest Bench	○	×	◎	3

一旦無視

参考：IF②

項目	説明
コンセプト	Test Bench側に持たせざるを得ない



実現系：自動テスト実装の戦略（自動テストアーキテクチャ以外）

VERISERVE

システムテスト自動化標準ガイドを参考に、戦略を一般的に評価した

項目	章番号	戦略
スクリプティングの技法	3.2	構造化スクリプト、共有スクリプト。件数が多くないのでキーワード駆動はしていない。また、パラメータ振りなどもないためデータ駆動もしていない。今後検討
比較タイミング	4.3/4.4	動的比較。リアルタイムに比較ができるシステムを採用したため。実施後比較が必要になったら検討するという位置付け
比較の複雑さ	4.5/4.6	複雑な比較。完全一致ではなく、比較すべきデータを抽出し、比較を行っているため
テストの感度	4.7	センシティブな比較。ATの規定が細かいため、センシティブになっている。目的がスモークテストなどになれば、判定を減らしてロバストにすることも検討可
前後処理の自動化	6	テスト実行そのものの前提条件作成などは自動化済み

アジェンダ

1. AUTOSARとは？
2. AUTOSAR Acceptance Test（AT）とは？
3. ATの自動化
4. 考察と今後の課題

考察

項目	考察内容
ATの自動化そのもの	複数回実施する、という意味ではどんな実装方法でもメリットが出そう。ROIの計測は必要
自動テストアーキテクチャ	現状規定のATのみ、ということであれば「2．簡易判定をTest Benchに」の戦略で問題なし。ただし、今回無視したスクリプト数を外すと、「4．全てTest Bench」という戦略が選ばれる可能性がある（次ページに再掲）
実装戦略（自動テストアーキテクチャ除く）	汎用性を高めなくてよかったため、キーワード駆動を使用しない、また、センシティブな比較を採用、とした。ターゲットが変わるとここの戦略も変化する

ATのアーキテクチャまとめ（再掲）

番号	コンセプト	スクリプト数	TCPの複雑さ	機能性	点数
1	全てSWC	×	◎	×	3
2	簡易判定をTest Benchに	×	◎	△	4
3	全判定をTest Benchに	×	△	○	3
4	全てTest Bench	○	×	◎	5

今後の課題

項目	課題内容
ATの拡張	ATのコンセプトに従い、機能深掘り、機能横展開が実施される際に、保守性が重要
AT外の自動テスト	今回はATとして実装したが、その他のテストレベルにも使用可能。その場合、アーキテクチャ、戦略の再考が必要
ECUとしての自動テスト	今回はIFとしてSWC＋Test Benchとして実装したが、Test Benchのみ切り出すことでECUとしての自動テストにも同じアーキテクチャが使用できる想定
テスト設計、実装の自動化	テストケースの生成（テスト設計）や、テストウエアの生成（テスト実装）の自動化を検討

ご清聴ありがとうございました。