

USDMを用いた 非機能要求の抽出

(株) デンソー
水藤 倫彰

2016/5/27(金) 14:10 – 14:50 横浜市開港記念会館

1. 背景
2. 課題
3. 提案手法
4. 結果
5. 考察
6. 今後の課題
7. まとめ

- 開発製品
 - 車載メータ用ソフト
 - 組み込みソフト
 - C言語を利用
- プロジェクト
 - LED駆動モジュール
 - 多くの意匠に関係
 - ミドルウェア
 - 多くのメータに利用

車載メータ



引用:Car Watch

影響箇所

多くの意匠、メータに影響するため、品質が特に重要

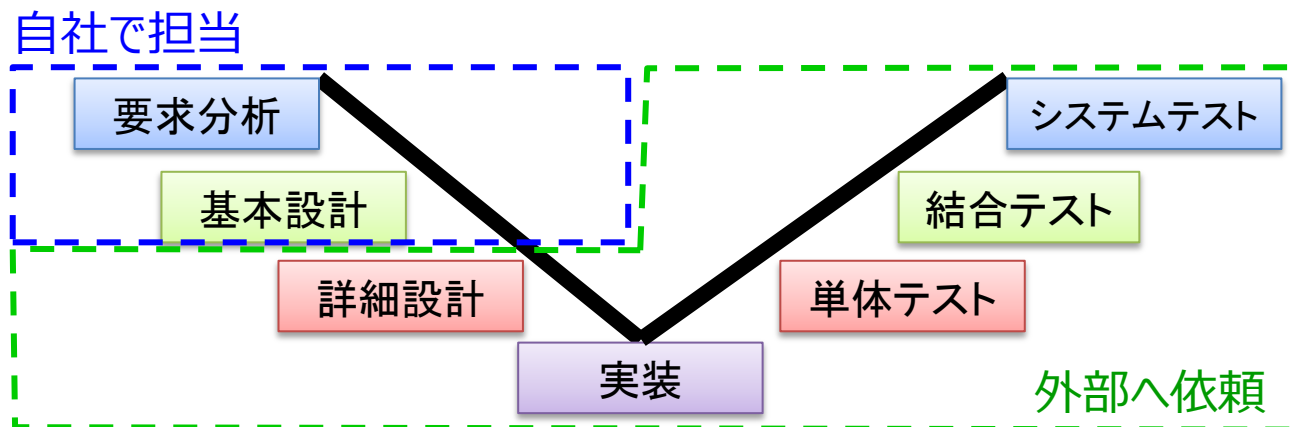
- LED駆動モジュール開発プロジェクト

- 派生開発

- 既存のコードに対して
リファクタリング + 新規要求に対応
- 既存コードから機能仕様書を作成

- 開発の一部を外部へ依頼

- 外部の成果物によってソフトウェアの品質が決定

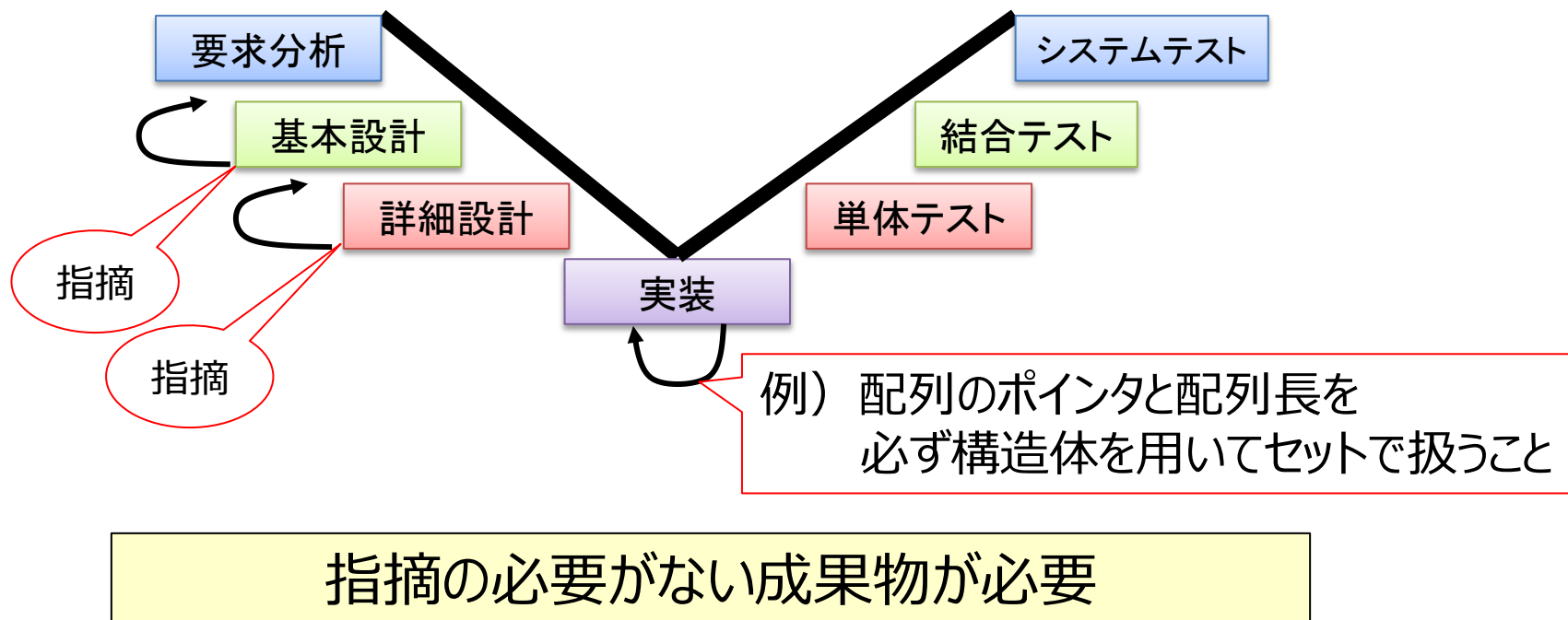


各工程の成果物はレビューにて品質を担保

• プロジェクトの問題

- 各レビューでソフトウェアの作り方に対する指摘が多発
- 対応にはソフトウェアを作り直す必要があり、大きな手戻りが発生

➡ 計画に対して大きな遅延が発生



- レビュー指摘多発の原因

- ソフトウェアの作り方に対する指摘
- 非機能要求に分類

➡ 設計、実装工程において
非機能要求の“入力”に不備

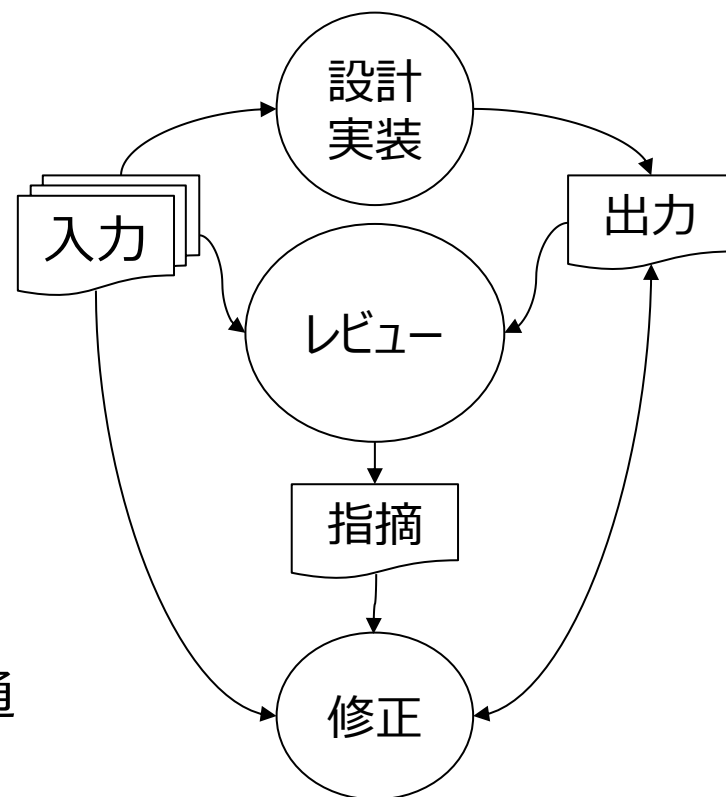
- 設計、実装工程の“入力”

- チェックシート

- メータ開発において各プロジェクトで共通
- 製品について一般的な項目のみ

➡ プロジェクトに必要な項目が抜けていた

プロジェクトに適した非機能の要求仕様書が必要



- 現場における非機能要求の現状

- 時間が確保できず、経験者のノウハウに頼っている

➡ 外部へ依頼した場合、暗黙知は伝わらない

- 過去起こった問題のみ確認

➡ 問題を未然に防ぐことは困難

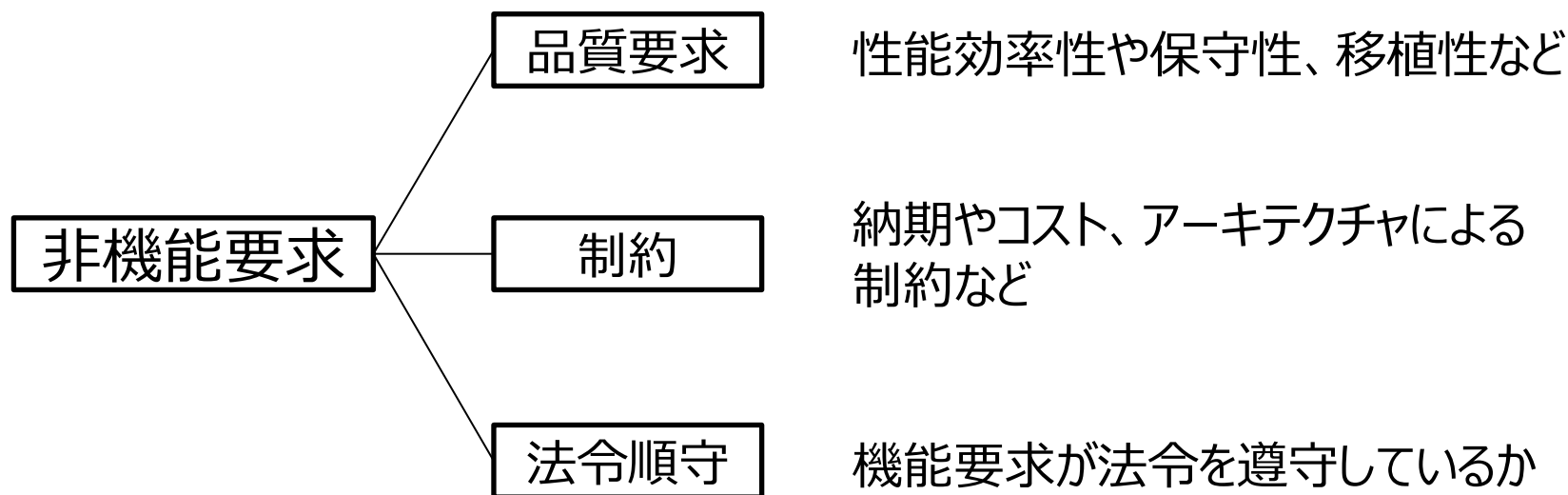
- 大量の確認項目を実施

➡ 1つ1つの項目確認が疎かになる

チェックシート

現場で実施可能な非機能要求の文書化方法が必要
(プロジェクトに適した、時間のかからない、容易な方法)

- 定義
 - 機能要求を満たす上での特性
- 分類



担当するプロジェクトでは品質要求が特に重要

- “前プロジェクト” のレビュー指摘を活用

- 非機能要求は流用できる可能性が高い

- 機能要求を満たす上での特性

- 同様の特徴を持つプロジェクトに流用可能



前プロジェクトと同様のレビュー指摘を抑えることが可能

- USDMの枠組みを利用

- 要求と仕様に階層化されているため、現場での活用が容易

派生開発に着目した方法で非機能の要求仕様書を作成

- 漏れなく要求仕様を導出する枠組み
- 要求仕様を“要求”と“仕様”に階層化
 - 要求により実現範囲を示す
 - 要求に含まれる仕様を漏れなく導出する枠組み

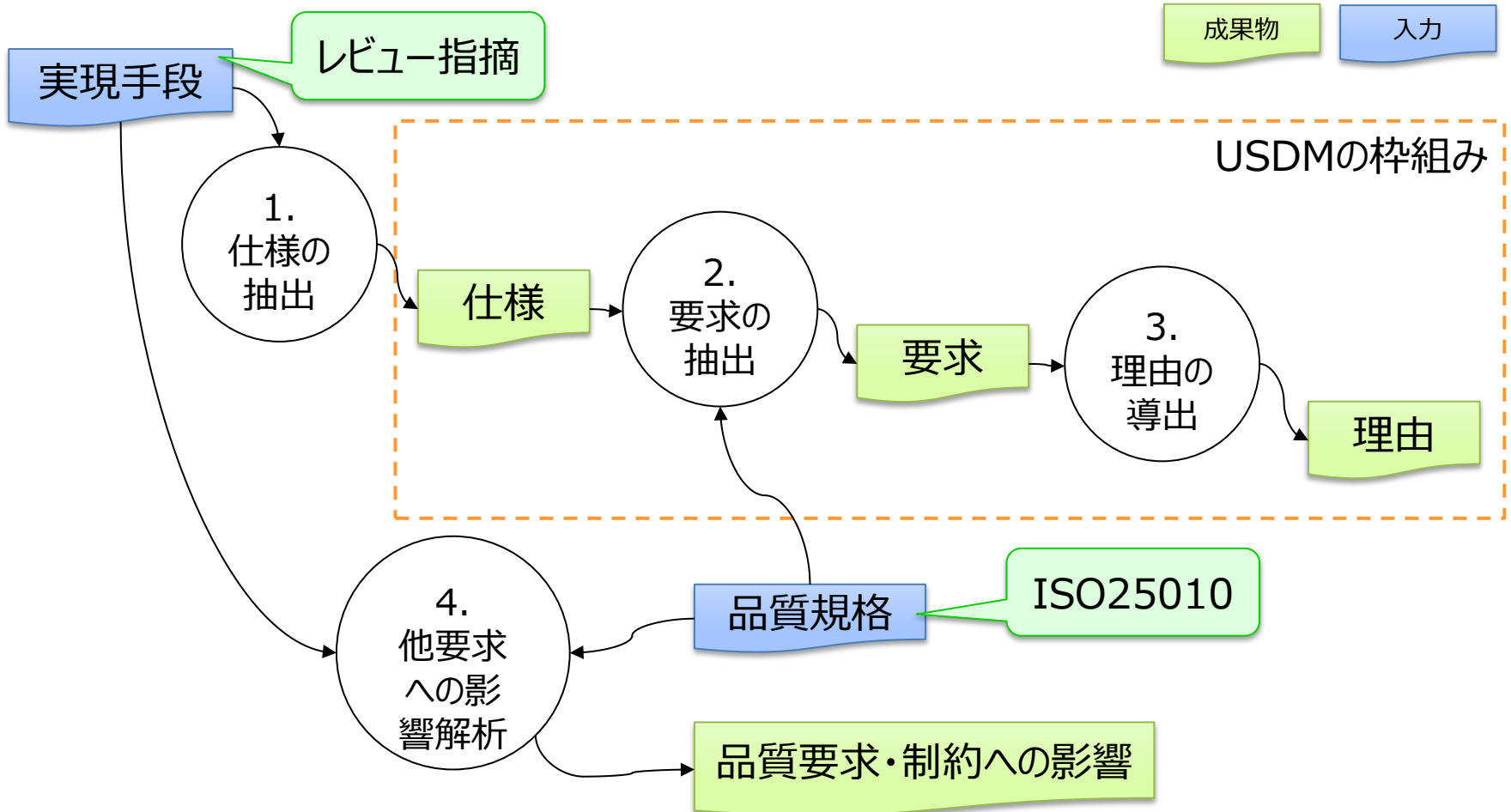
範囲を示すことで、範囲に含まれる仕様を漏れなく導出

要求	FND07	検索結果を扱いやすく表示し、
	理由	目的のメールが1つとは限らないので、絞り込めるような操作
<input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/>	FND07-01	検索されたメールの件数を一覧の上に表示する
<input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/>	FND07-02	該当するメールが存在しない時は「該当なし」を表示する
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	FND07-03	検索されたメールの「Subject」を一覧で見せる

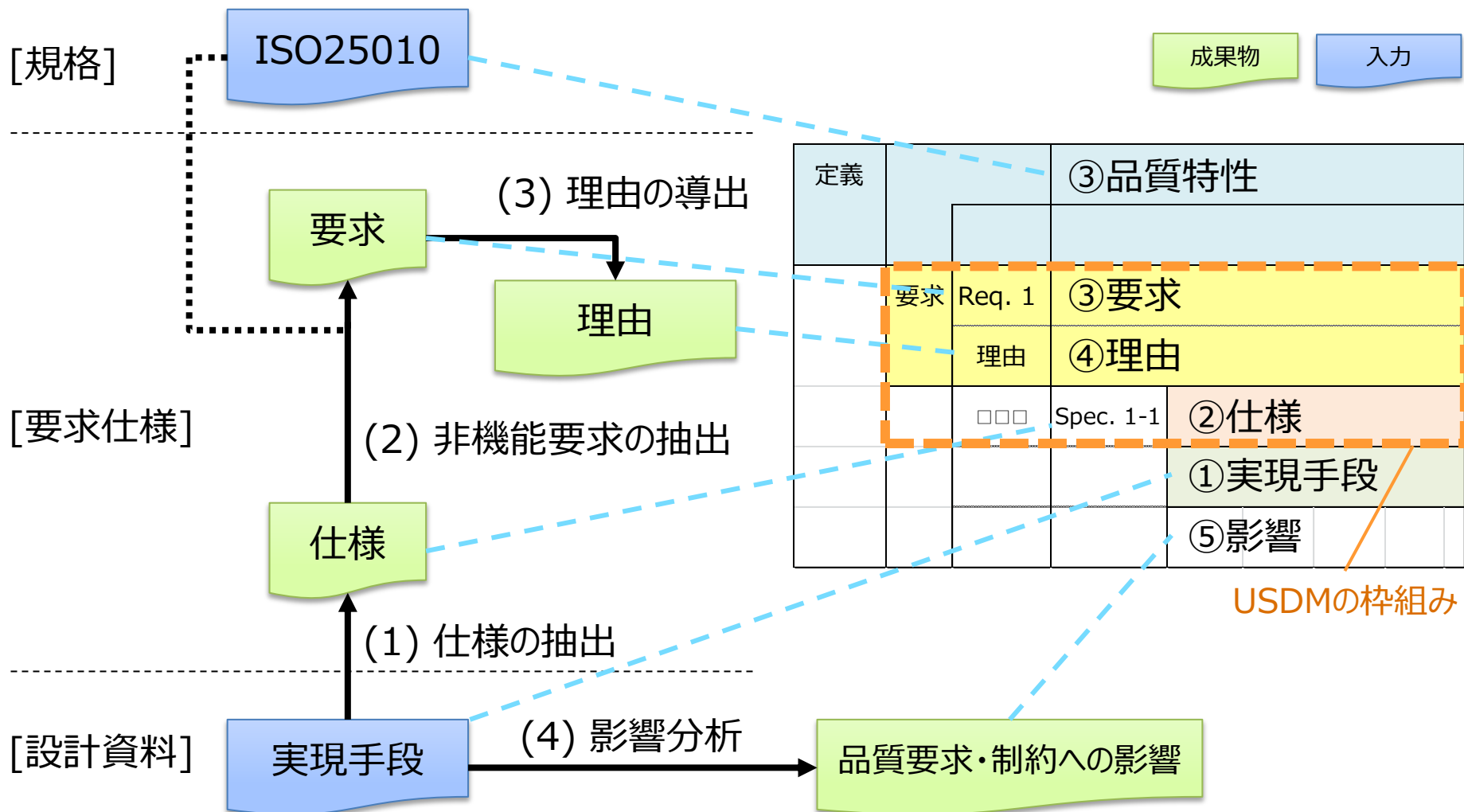
要求レベルでプロジェクトへの適性を判断可能

出典: 要求を仕様化する技術・表現する技術, 清水 吉男, 技術評論社

- レビュー指摘から非機能要求仕様を作成

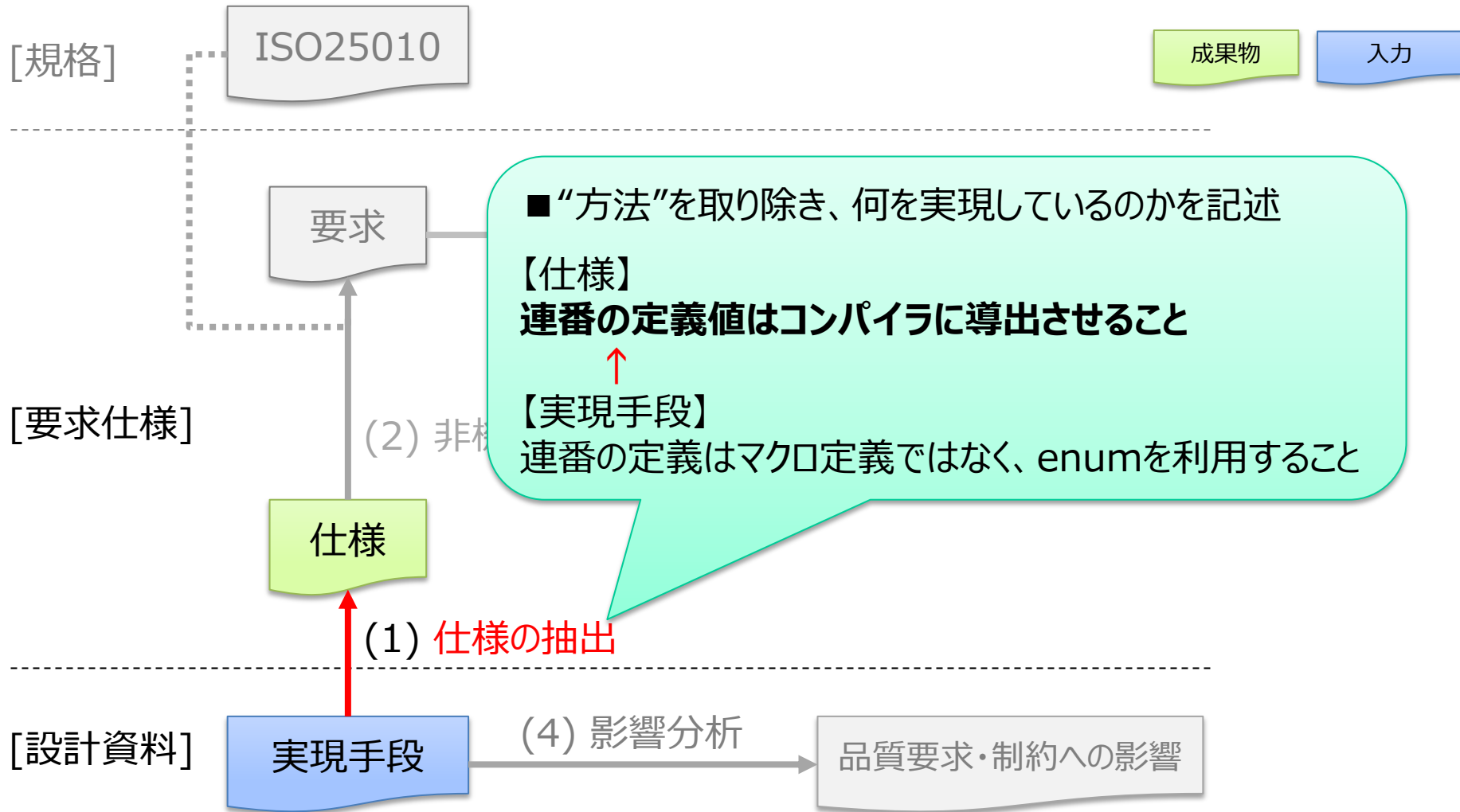


3. 提案手法

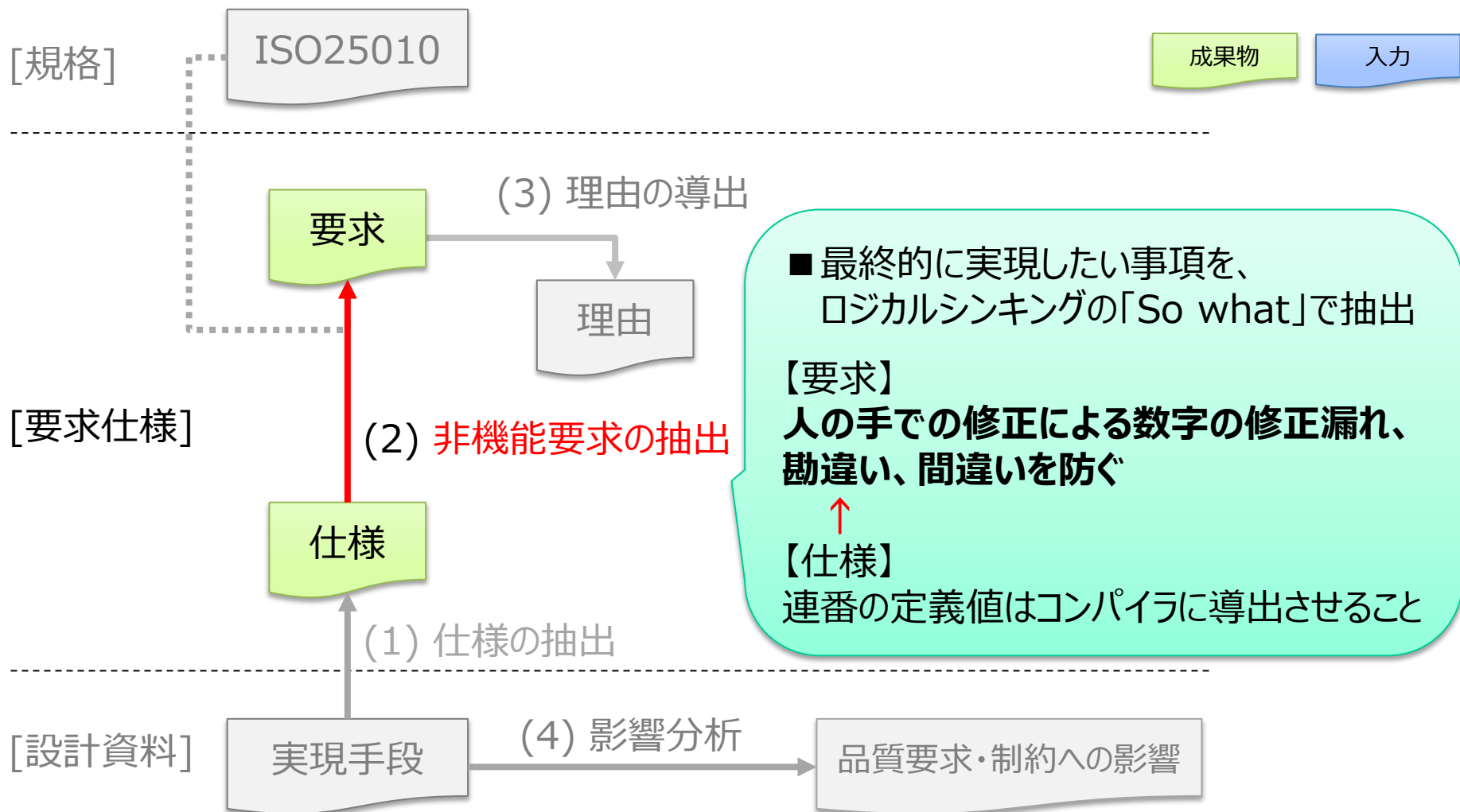


USDМの枠組みに、実現手段, 品質特性, 影響を記入する欄を追加

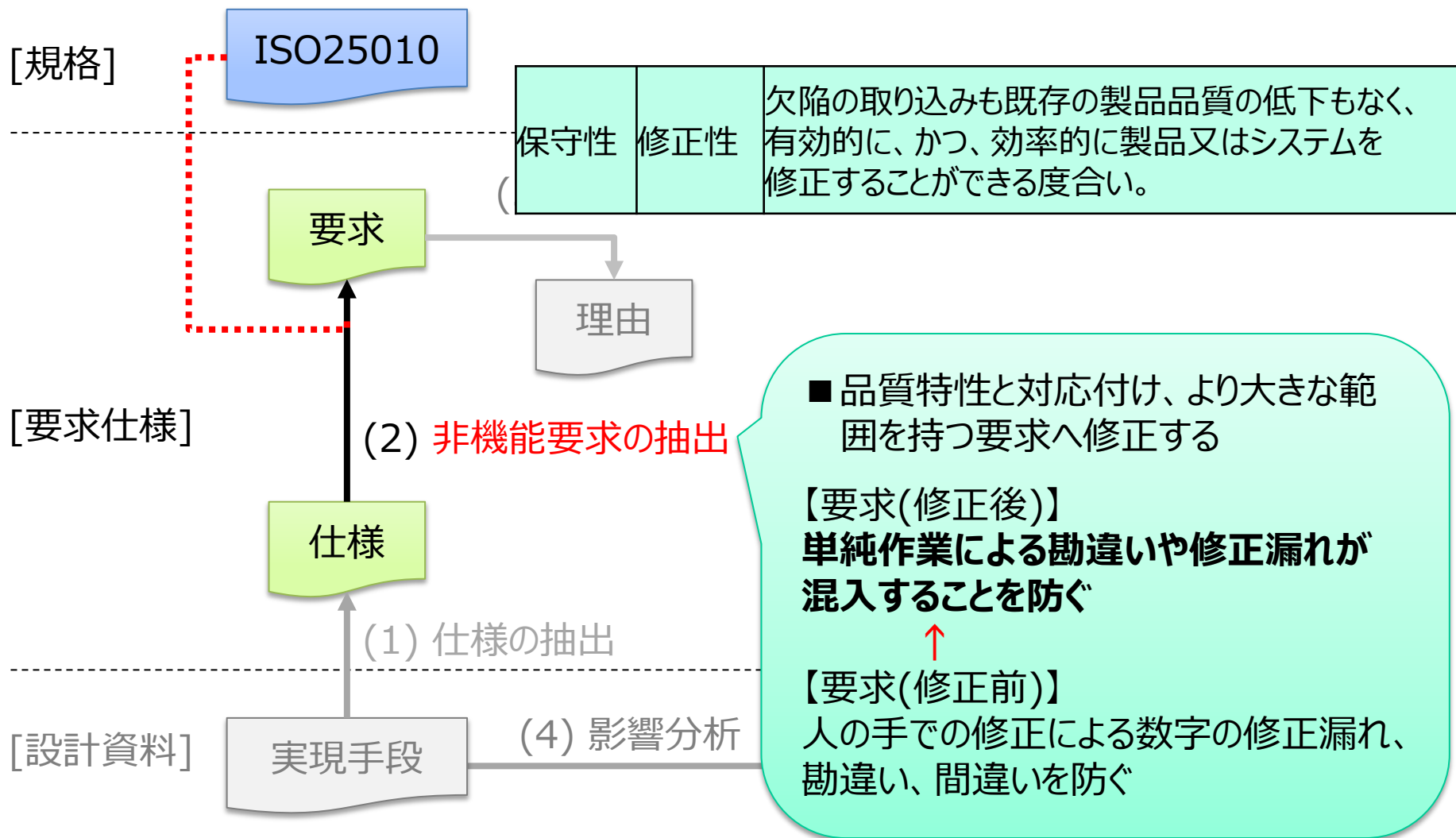
3. 提案手法：仕様の抽出



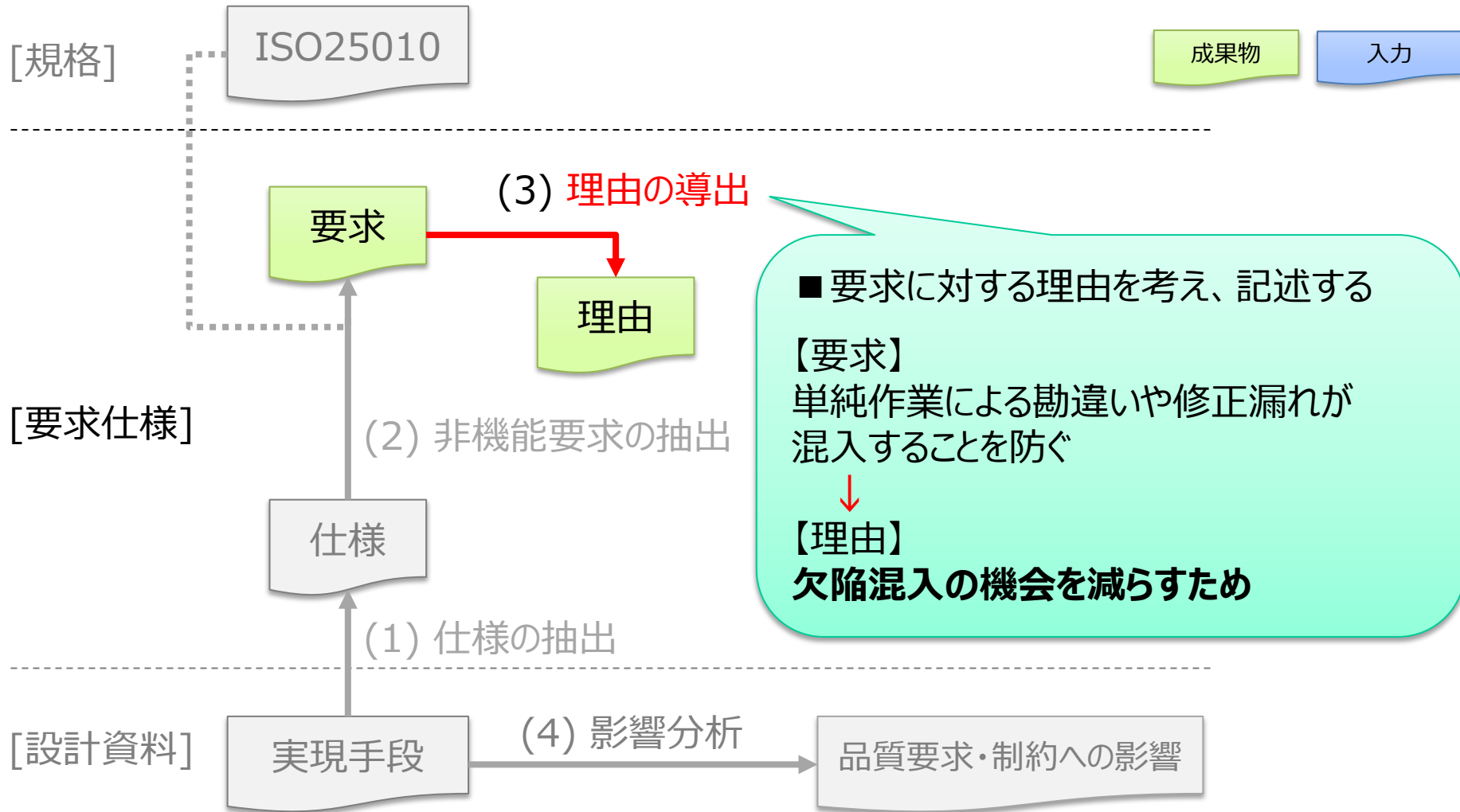
3. 提案手法：非機能要求の抽出



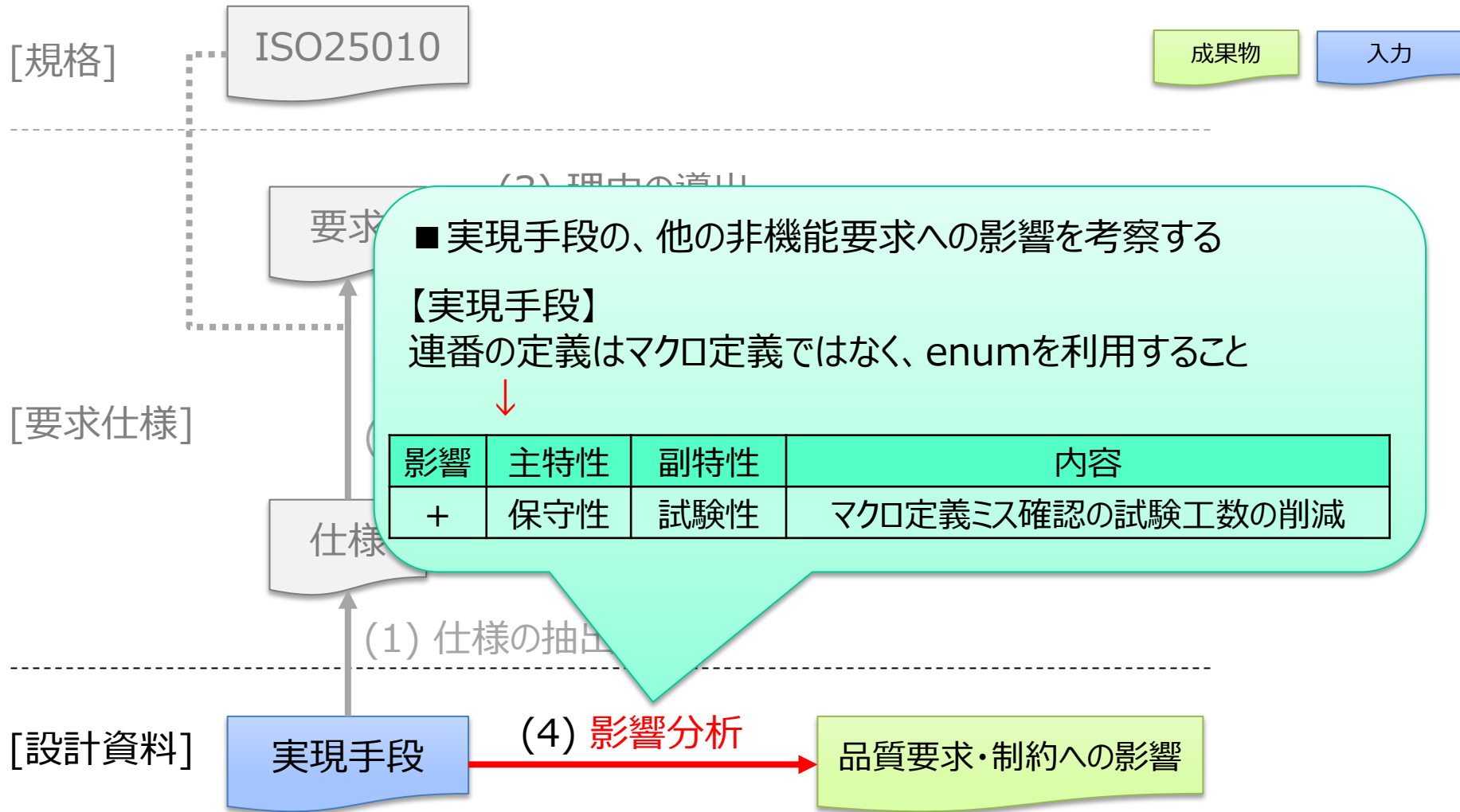
3. 提案手法：非機能要求の抽出



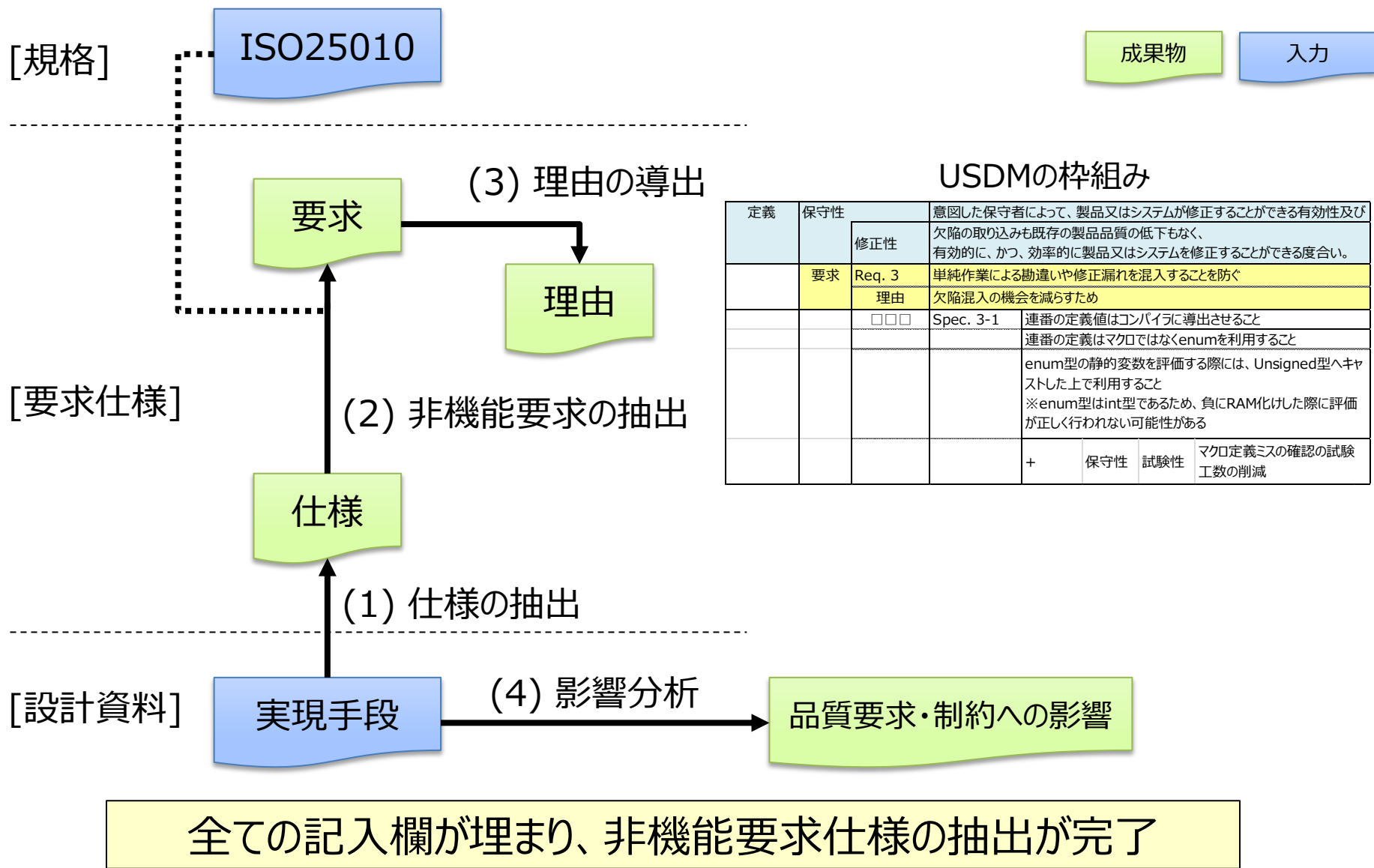
3. 提案手法：理由の導出



3. 提案手法：影響分析



3. 提案手法



- LED駆動モジュール開発プロジェクト
 - 実現手段として非機能に関するレビュー指摘25件を利用

定義	保守性		意図した保守者によって、製品又はシステムが修正することができる有効性及び効率性の度合い。	
	修正性		欠陥の取り込みも既存の製品品質の低下もなく、有効的に、かつ、効率的に製品又はシステムを修正することができる度合い。	
要求	Req. 3		単純作業による勘違いや修正漏れを混入することを防ぐ	
	理由		欠陥混入の機会を減らすため	
	□□□	Spec. 3-1	連番の定義値はコンパイラに導出させること	仕様
			連番の定義はマクロではなくenumを利用すること	
			enum型の静的変数を評価する際には、Unsigned型へキャストした上で利用すること ※enum型はint型であるため、負にRAM化けした際に評価が正しく行われない可能性がある	
			+ 保守性 試験性 マクロ定義ミスの確認の試験工数の削減	影響
	□□□	Spec. 3-2	配列・構造体のサイズはコンパイラに導出させること	
			配列・構造体のサイズを求める際は、sizeof演算子を利用すること スコープ外の配列サイズを求める際に、sizeofを利用しないこと 【説明】 配列のサイズ情報は配列を定義したスコープ内のみ利用可能なため	
			+ 保守性 試験性 配列サイズの確認の試験工数の削減	

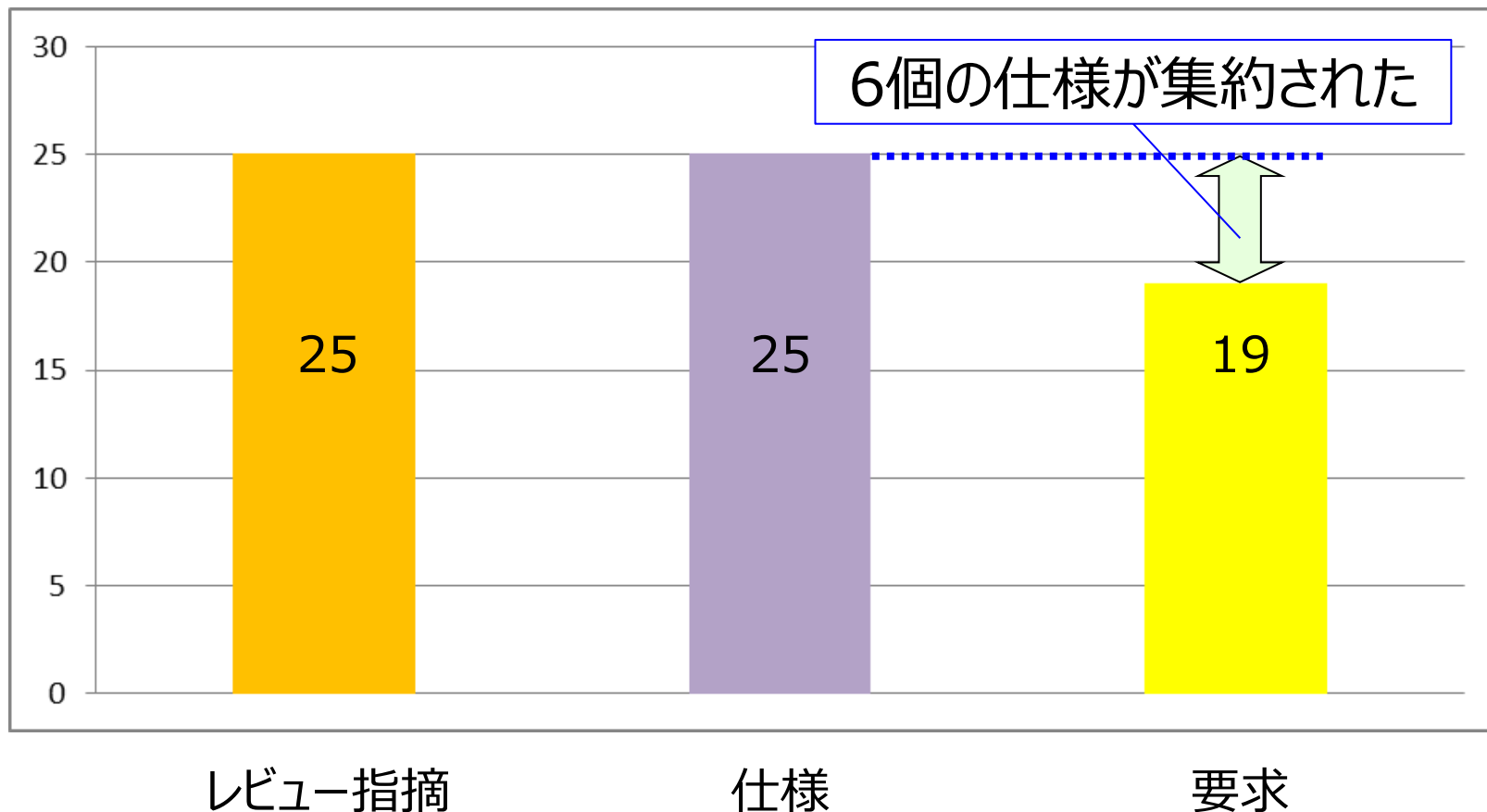
レビュー指摘

仕様

影響

プロジェクトにおける非機能の要求仕様を文書化

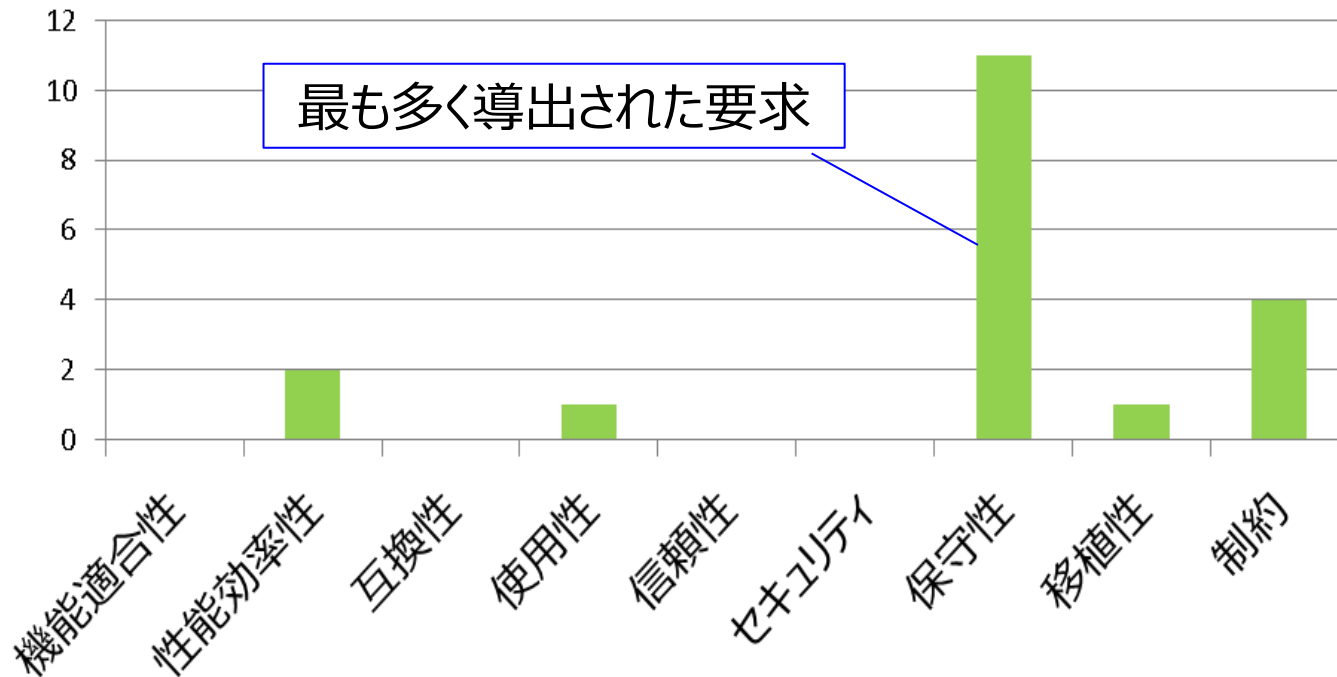
- 項目数の変化



項目数の削減により要求仕様の活用が容易に

- 非機能要求の分類

- 品質特性の主特性、及び制約に分類



担当する開発プロジェクトは保守性を重視

プロジェクトの特徴を分析可能

● 提案手法の評価

－ 課題

- 現場で実施可能な非機能の要求仕様書作成方法が必要
(プロジェクトに適した、時間のかからない、容易な方法)

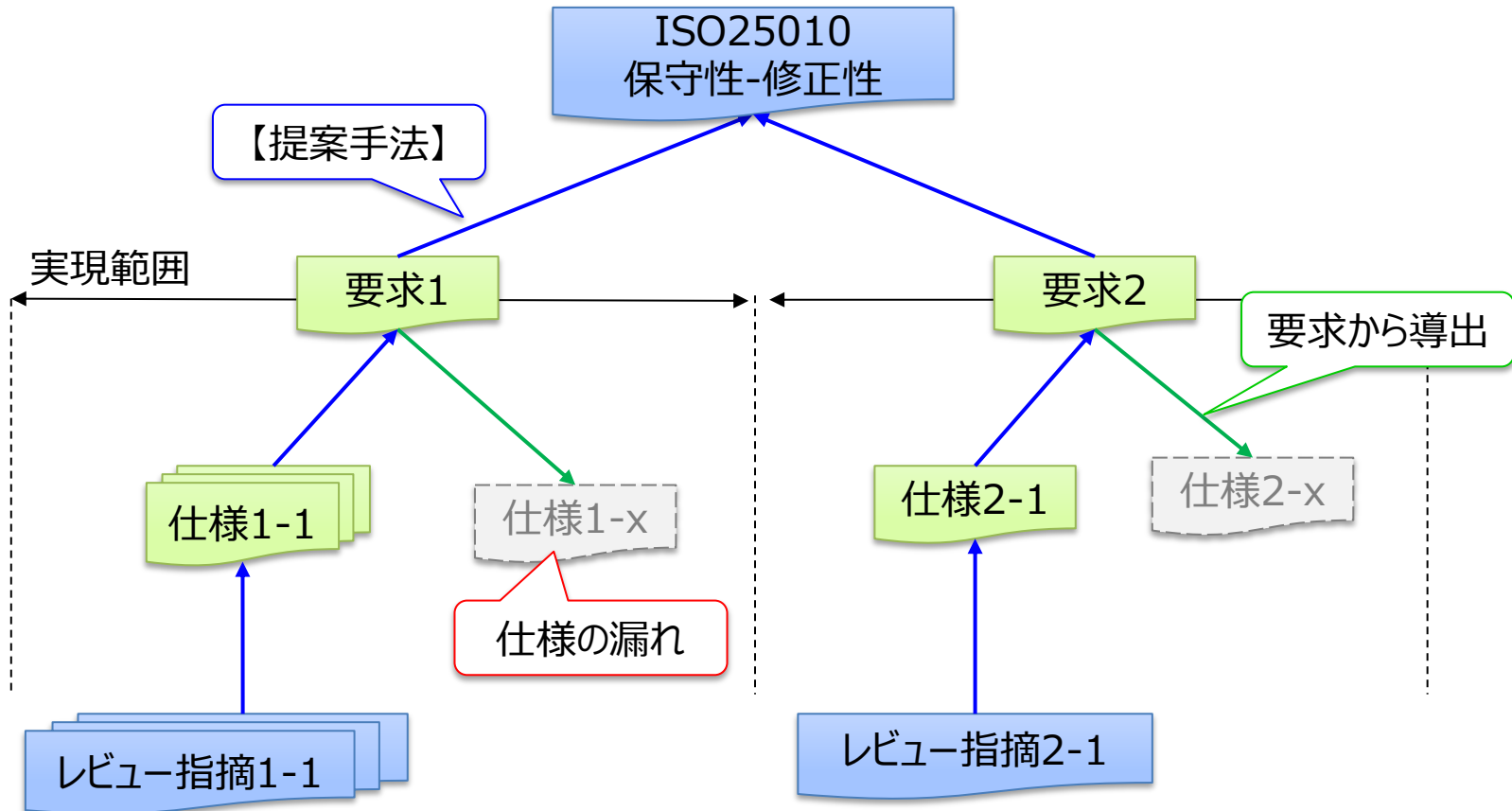
－ 評価

	要求仕様の作成	要求仕様の選択
新規に考えないため 時間も早く、容易	現在のプロジェクト	次のプロジェクト
実行時間	◎	○
実行可能性	◎	○
仕様の網羅性	△	△

選択肢にない仕様は漏れる

要求仕様を蓄積することにより、仕様の漏れに対応

- 非機能要求の仕様漏れへの対策検討



要求の表す実現範囲内の仕様を導出することで防ぐことが可能

- プロジェクトでの試行
 - 非機能要求仕様書の活用
 - 利用時の使いやすさを考えたフォーマットの作成
 - 利用時のワークフローの定義
 - 定量的評価を実施

- 非機能要求における制約の分類について調査、検討

- 背景
 - 非機能に関するレビュー指摘から大きな手戻りが多発
- 課題
 - 現場で実施可能な非機能要求の文書化方法が必要
(プロジェクトに適した、時間のかからない、容易な方法)
- 提案手法
 - 前プロジェクトのレビュー指摘から非機能の要求仕様を作成
 - USDMの枠組みを利用
- 結果
 - 同じ特性を持つプロジェクトで有効な要求仕様を作成
 - 品質規格を利用することでプロジェクトの特性を分析可能

DENSO