

派生開発における協力会社への 前提知識の伝達方法

- 目次
1. 研究経緯
 2. 現状分析
 3. 解決策
 4. 検証
 5. まとめ

第6分科会Aグループ

リーダー	伊藤 敏也	(株式会社インテック)
	羽瀨 喜英	(株式会社インテック)
	鎌田 朋洋	(アンリツエンジニアリング株式会社)
	山本 貴之	(アンリツエンジニアリング株式会社)

■ 派生開発の現場で困っていること

今まで付き合いのない協力会社との協業で不具合が多く発生してしまう

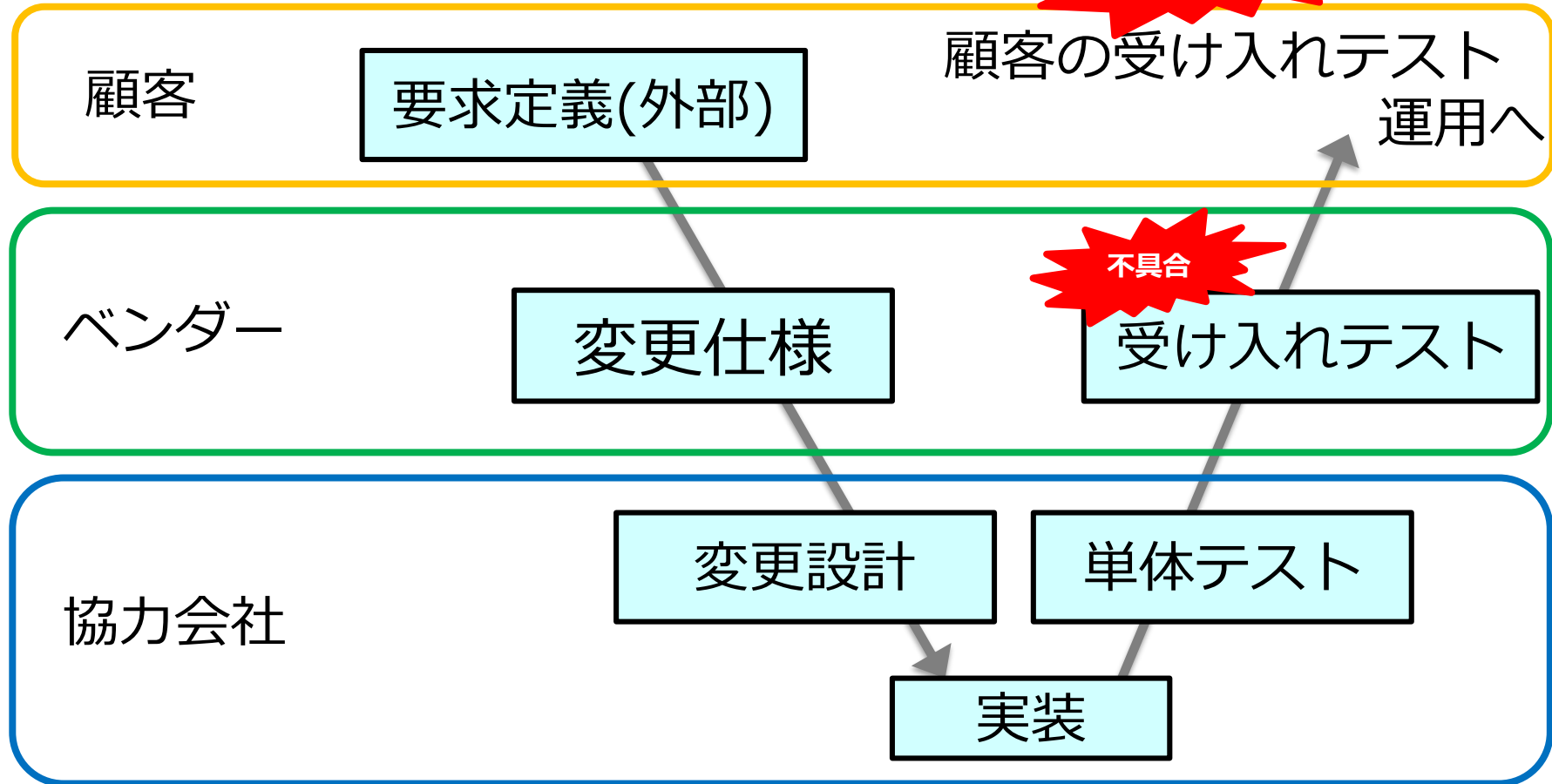
Q:そもそも、なぜ付き合いのない協力会社との協業を迫られるのか？

- ・ 会社全体の方針としてのコスト削減の要求
- ・ 案件増加によるリソース不足

担当者の立場ではチーム編成に関われないので、まず現状を分析した

■ 問題はどこで発生したか

不具合



**顧客側で不具合が表出すると大問題！
⇒ 協力会社起因の不具合を分析した**

■不具合の原因内訳

実装誤り

単純ミスや実装技術不足

12件
20%

その他

2件
3%

知識不足

調査・設計といった各作業の前提知識が欠落

18件
30%

影響調査不足

前提知識を持つが、追加・変更内容の影響範囲を特定できず不具合が発生

12件
20%

設計不備

前提知識を持つが設計誤り、パターン漏れ、変更漏れが発生

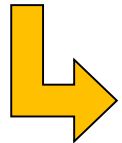
16件
27%

不具合の原因件数に大差はない

⇒費用対効果の視点で解決項目をピックアップ

■費用対効果の視点で考えると

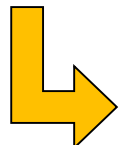
- ・設計不備、影響調査不足、実装誤りの解決策
作業プロセスの見直し、開発担当者のスキルアップ



- ・委託側からは制御が困難
- ・時間、コスト増

- ・知識不足起因の不具合の解決策

製品に関する前提知識の提供

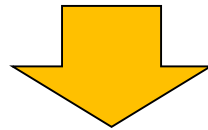


- ・初期コスト(伝達する知識の選別)を要するが容易

比較的容易に実施でき、
短期間で改善につながる知識不足を解決する

■ 具体的な知識不足起因の不具合とは...

- ① ソースコード中の“Antenna”変数に2つの解釈があることを知らずにコードを移植し、不具合発生
- ② ある機能の単体テストの際は、テスト用にコードの修正が必要であったが、それを知らなかったため正しくテストができていなかった



**用語、開発環境、テーブル構造、性能要件
といった前提知識を知ってさえすれば
不具合の混入を防げたのでは？**

何故、この知識が伝達できていないのか？

■不足していた前提知識の分析

表：伝わらなかった前提知識の記載状況

ドキュメントへの記載状況	件数
記載がない（欠落）	10
記載はあるが見つつけづらい（過多）	4
記載内容がわかりにくい	4

不足している前提知識を補い、
情報過多にならないように伝える必要がある

■ 解決の方針

既存のChipsに

- ・ 記載がない (欠落)
- ・ 記載内容がわかりにくい

不足している知識を補うために新しいChipsを作成



Chips
(前提知識をまとめたドキュメント)

**ドキュメントを増やしすぎると、
情報過多となり必要な情報が見つけづらい！**

■ 解決策の提案

『機能一覧Chips紐付け表』

	課題	解決策
1	前提知識(Chips)が「見つけづらい」	機能一覧Chips紐付け表によるChipsの伝達
2	前提知識(Chips)に「記載がない」または、「わかりにくい」	Chipsの追加と機能一覧Chips紐付け表のメンテナンス
3	前提知識の伝達方法	「機能一覧Chips紐付け表」の使用による改善手順

**機能一覧Chips紐付け表を利用することで、
成果物の品質向上が期待できる！**

■ 機能一覧Chips紐付け表 (1)

『機能一覧Chips紐付け表 (マスター) 』

機能	Chips					
	テーブル関連図.xlsx	DB参照規約 (コーディング標準書.xlsx 4.1章)	DB更新規約 (コーディング標準書.xlsx 4.2章)	Web画面作成規約 (コーディング標準書.xlsx 3章)
ログイン		○		○		
条件検索	○	○		○		
検索結果一覧	○	○		○		
商品情報詳細	○	○		○		
商品情報ダウンロード	○	○		○		
商品情報Webメンテナンス		○	○	○		
商品情報ファイルアップロード			○	○		
画像登録ポップアップ			○	○		
メールングリスト登録			○	○		
...						
...						

**各機能ごとに派生開発時に参照すべき
Chipsの紐付けを行う**

■ 機能一覧Chips紐付け表 (2)

『機能一覧Chips紐付け表 (テータリング済み)』

機能	派生開発要求			Chips					
	表示項目の 入れ替え	参照機能の HTML5対応	商品登録チェッ ク修正	テーブル 関連図.xlsx	DB参照規約 (コーディング 標準書.xlsx 4.1章)	DB更新規約 (コーディング 標準書.xlsx 4.2章)	Web画面作成規約 (コーディング 標準書.xlsx 3章)
ログイン		○			○		◎		
条件検索		○		○	○		◎		
検索結果一覧		○		○	○		◎		
商品情報詳細	○	○		○	◎		◎		
商品情報ダウンロード				○	○		○		
商品情報Webメンテナンス	○		○		◎	◎	◎		
商品情報ファイルアップロード			○			◎	○		
画像登録ポップアップ			○			◎	○		
メーリングリスト登録						○	○		
...									
...									

要求と改修機能の紐付けを行い、
今回の開発に必要なChipsの精査を行う

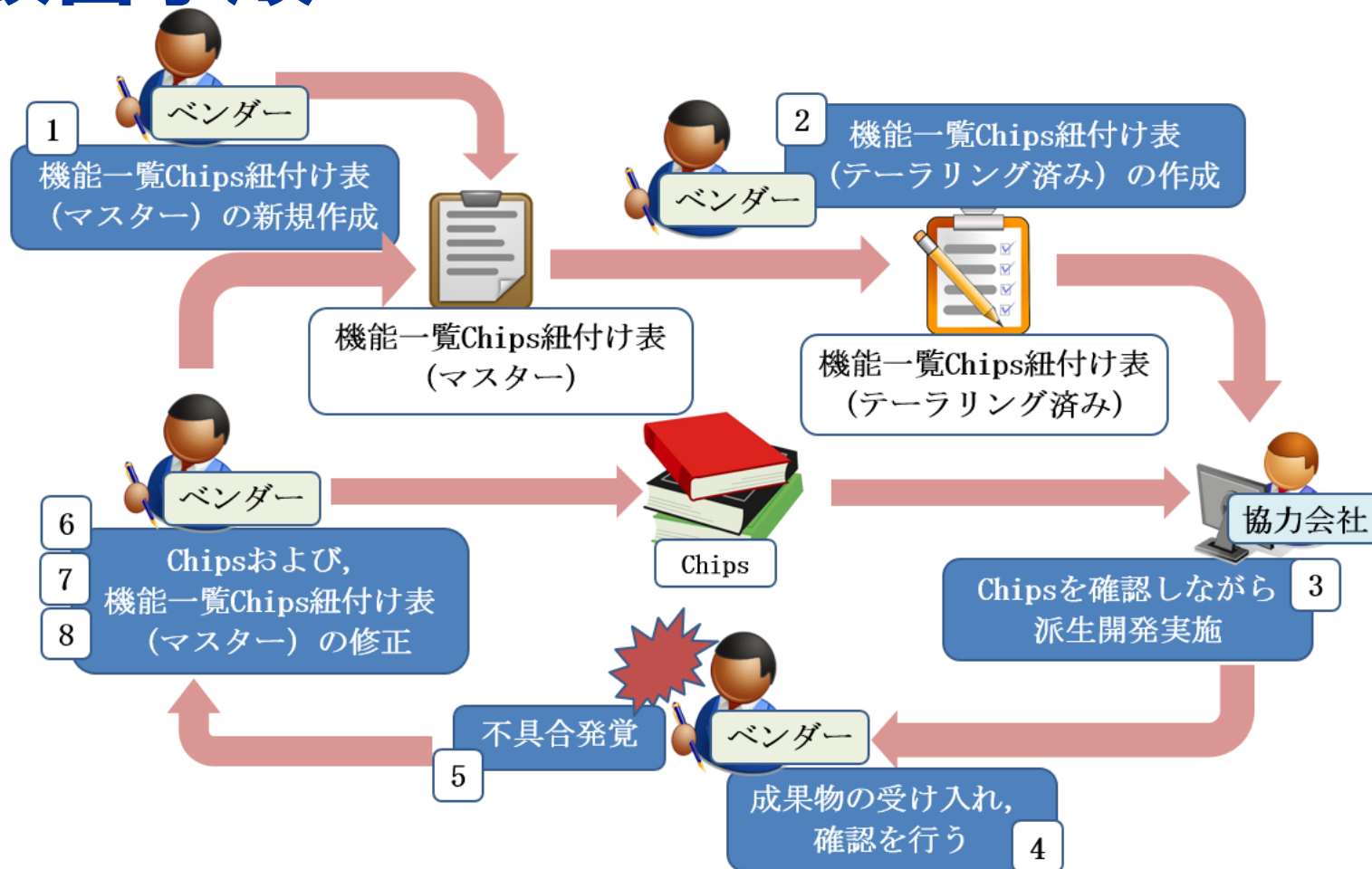
■メンテナンス方法 (1)

No	タイミング	内容
1	機能の追加時	新しい機能を機能一覧Chips紐付け表（マスター）に追記し、それに紐づくChipsに○印を記載する。
2	機能の改修時	改修された機能に紐づくChipsを新たに追加、削除を行う。
3	不具合の分析時	3-1[記載がない] 新たに記載がないChipsを作成し、機能との紐付けを行う。

■ メンテナンス方法 (2)

No	タイミング	内容
3	不具合の 分析時	3-2[見つけづらい] Chipsが記載されていない場合は追記し、機能との紐付けを行う。Chipsが記載されている場合は、紐付けの確認を行う。
		3-3[わかりにくい] 既存のドキュメントを修正し、他に同様の記述がないか横展開チェックを行ってChipsを改善する。
4	協力会社からの問い合わせ発生時	協力会社から問い合わせがあった場合もNo3と同じように分析を行い、3-1~3-3と同様の対応を行う。

■ 改善手順



**サイクルを繰り返していくことで
情報伝達を改善できる！**

■ 検証内容

以下2点のシミュレーションを実施

- ・ 過去不具合(60件)のうち**協力会社起因**の不具合は減るか？

機能一覧Chips紐付け表で不具合を防げるか協力会社の開発者に確認を取る

- ・ 機能一覧Chips紐付け表の**作成・メンテナンス工数**はどれくらいか？

初期作成時、案件発生時、機能およびChips追加時の工数を見積り

不具合を防げるか？工数はどれくらいか？

■ 不具合の削減

分類	解決の可否	件数
記載はあるが見つげづらい (過多)	解決可能	4
わかりにくい	過去の不具合一覧を元に対策するので一度目の発生を防げないが、二度目の発生を防ぐことができる	4
記載がない		10

■ メンテナンス時間

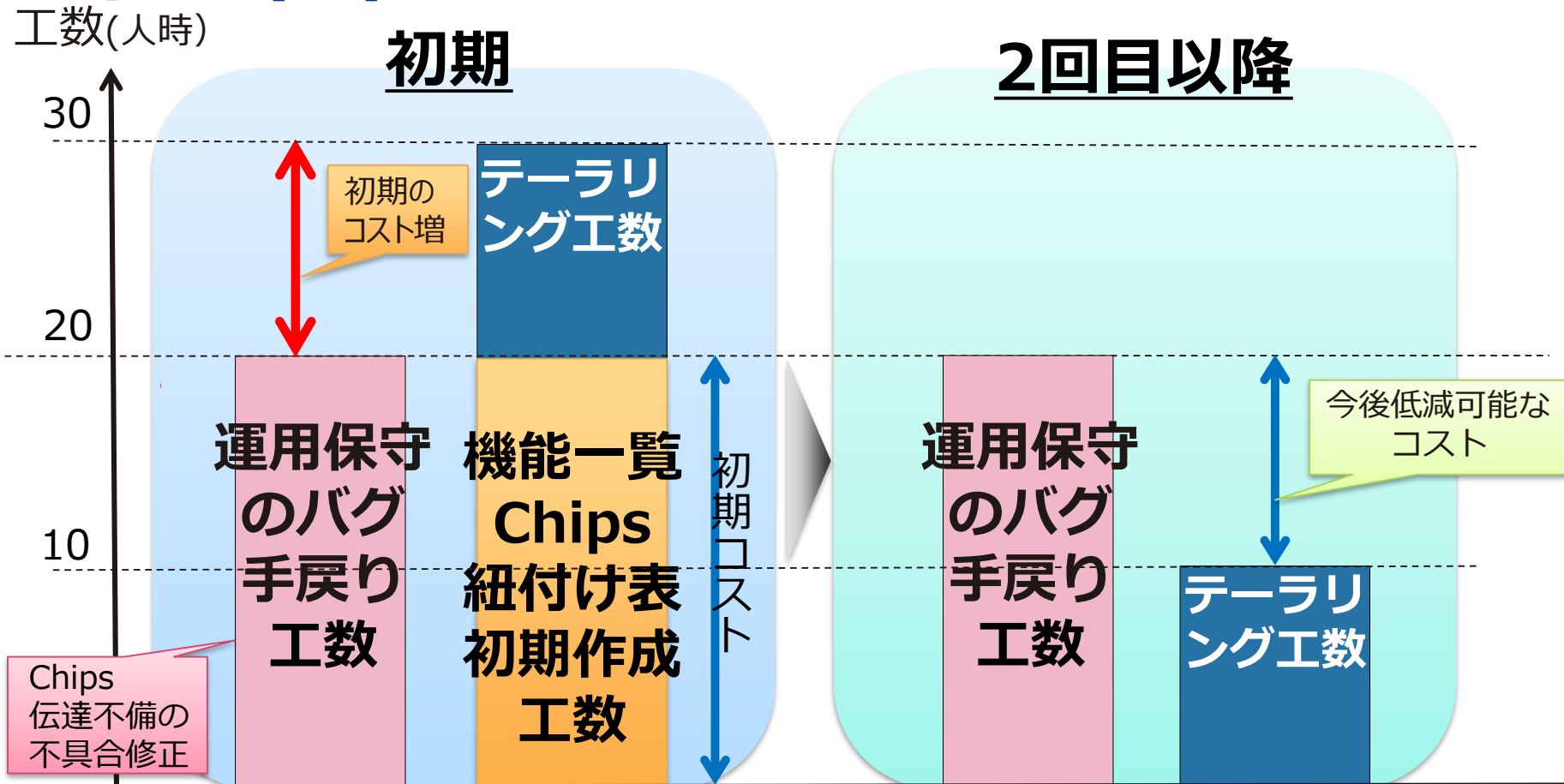
18件低減

機能一覧Chips紐付け表 (マスター)	作成工数	機能一覧Chips紐付け表 (テラリング済)	作成工数
初期作成工数 機能数:228 Chips数:50	20時間	案件毎テラリング工数 派生開発要求件数:15	10時間

↑
Chips作成工数は含まない

機能一覧Chips紐付け表により不具合を
18件低減、工数は初期20h、案件毎10h

■考察 (1) Chips伝達の不備が1案件で1件程度(20h)発生



**2回目以降、10時間の工数追加
Chips伝達不備はゼロに！**

■考察（2）

新規の協力会社の
学習容易化を実現

機能一覧Chips紐付け表
テーラリング工数

10時間

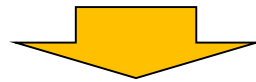
プログラム設計書作成工数
(見積)

4600時間
(約30人月)

完全に整備しても情報過多のため改造に
必要な情報にたどりつけない可能性も。。。

■成果

機能一覧Chips紐付け表を使えばベンダーと新規の協力会社の知識の差を埋めることができる。



協力会社に依頼した成果物の品質向上につながる。

■今後の課題

- 今回の施策がベンダーの効率を損なわず運用でき効果を得られるかを実際のプロジェクトで確認する。
- 知識不足に次ぐ設計不備についても問題解決を図る。

**新規の協力会社を開発を依頼する際の
品質を安定させ続ける**

ご清聴ありがとうございました
