ソフトウェアテストシンポジウム2014
セッションD2-1　事例発表

FUJITSU
shaping tomorrow with you

テスト設計のタイミングと手法の変更による、品質向上と生産性向上

ー順序を入れ替え、繰返し考えるー

株式会社富士通マーケティング
ソリューション事業本部　GLOVIA事業部　開発部
本部プロセスチーム
松浦豪一

1．自己紹介

FUJITSU

- 1986年入社
  汎用機ソフトウェア・ミドルウェアの開発
- 2001年からGLOVIA-C会計の開発
- 2005年からKAIZEN活動を実践
  →KAIZEN塾（FST1期）伝道師型
- 2006年11月より、富士通㈱に出向
- 2007年8月よりGLOVIA/MIの開発リーダ
- 2010年　富士通に転社し、富士通マーケティングに出向
  富士通のアジャイル支援チームとして活動
- 2011年　GLOVIA SUMMITの開発
- 2013年　本部プロセスチームとなり、CMMI推進役となる。

ソフトウェアテストシンポジウム2014
セッションD2-1　事例発表

FUJITSU
shaping tomorrow with you

2. 問題発生

２－１　チェック処理追加
２－２　処理内容
２－３　発生した問題
２－４　問題を検出するには

２－１　チェック処理追加

FUJITSU

- 振込先情報にチェック追加(未入力、銀行コード存在)

■ 振込先情報

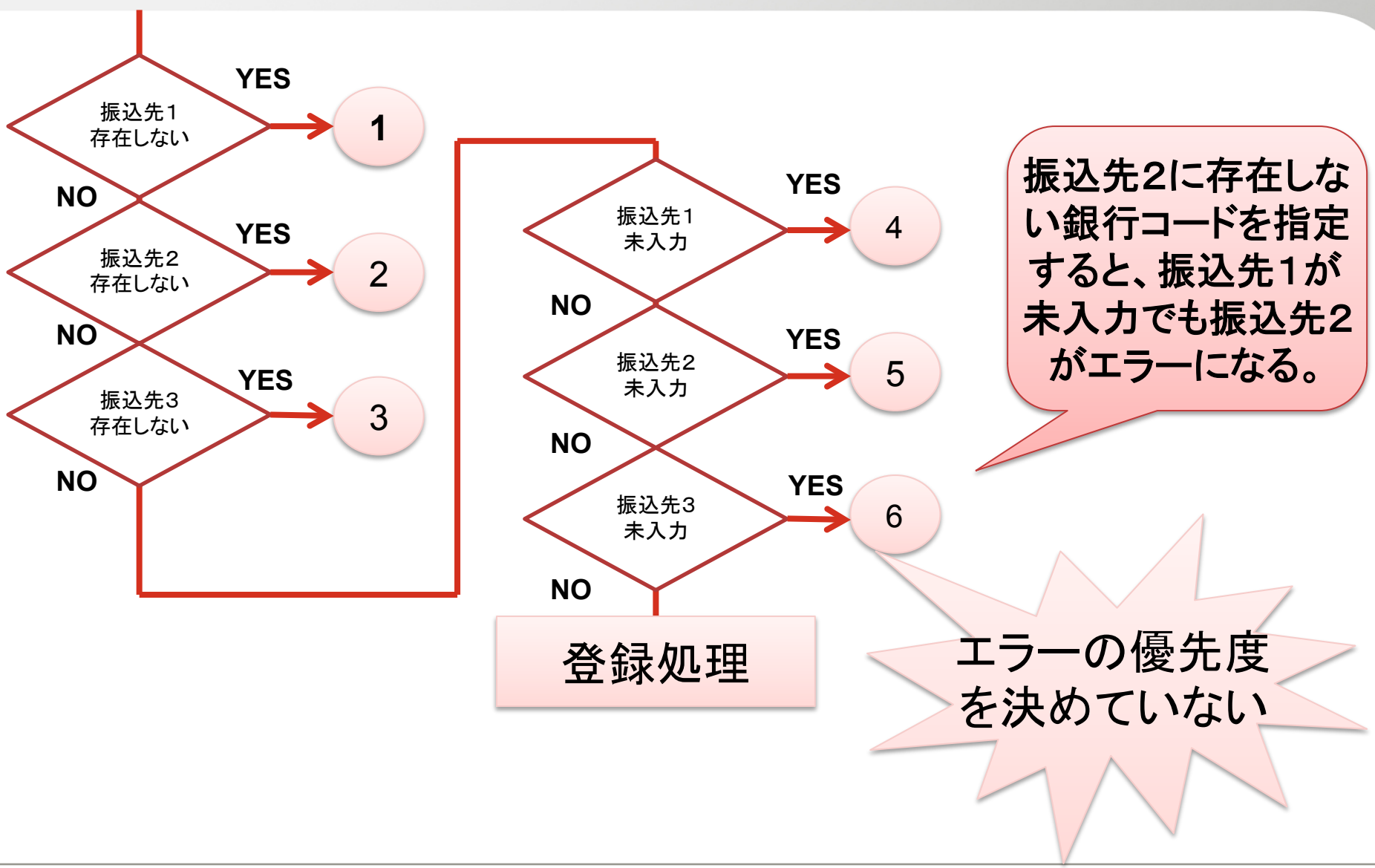
	振込先1		振込先2		振込先3	
銀行コード	0010	AAA銀行	0020	BBB銀行	0030	CCC銀行
店舗コード	111	東京支店	222	神田支店	333	秋葉原店
口座種別	普通預金		普通預金		普通預金	
口座番号	00000001		00000002		0000003	

■ 振込口座名義人

名義人	○○　○○
名義人(カナ)	マルマル　マルマル

登録

２－２　処理内容



２－３　発生した問題

振込先2の
銀行コードが
誤りです

■ 振込先情報

	振込先1		振込先2		振込先3	
銀行コード			1133		0030	CCC銀行
店舗コード			222		333	秋葉原店
口座種別			普通預金		普通預金	
口座番号			00000002		0000003	

■ 振込口座名義人

名義人	○○　○○
名義人(カナ)	マルマル　マルマル

登録

２－４　問題を検出するには

FUJITSU

- 全パターンをチェックするためには組み合わせで
  １２パターンやらないといけない。

要因		A	B	C	D
		金融機関コードエラー	振込先の未入力エラー		
因子	1	振込先１	振込先１		
	2	振込先２	振込先２		
	3	振込先３	振込先３		
	4	なし	なし		

組合せ条件（要因/因子の番号をカンマで区切って1条件/1行で記入。
記入例：要因Aと要因Bと要因Cを組合せる→A,B,C、要因Aの因子1,3と
1	A-1,B-2/3/4
2	A-2,B-1/3/4
3	A-3,B-1/2/4
4	A-4,B-1/2/3

振込先2コードエラー　かつ
　　振込先1未入力

振込先3コードエラー　かつ
　　振込先1未入力

振込先3コードエラー　かつ
　　振込先2未入力

- チェック内容をベースに仕様を検討するが、
  チェック内容の組み合わせでパターンは考えない。

ソフトウェアテストシンポジウム2014
セッションD2-1　事例発表

FUJITSU
shaping tomorrow with you

3. 改善策1　テスト設計を先にやる

３－１　順番を入れ換える
３－２　パターンを考える
３－３　効果
３－４　データと組み合わせ
３－５　何が起きているのか

３－１　順番を入れ替える

FUJITSU

テスト設計

プログラミング

３－２　パターンを考える

FUJITSU

- チェックするパターンを最小にする
  - A-1/2/3,B-1/2/3,C-1/2/3で設計しない
  - A-1/3で、振込先１のチェックが最優先
  - A-2,B-1/3で、振込先１→振込先２→振込先３の順

要因		A	B	C
		振込先１情報	振込先２情報	振込先３情報
因子	1	未入力	未入力	未入力
	2	正常	正常	正常
	3	エラー	エラー	エラー

３－３　効果

FUJITSU

- テスト仕様を決めることによって、設計の不十分な点を補完する。
- 組み合わせパターンが最小になるように設計する。
- 危険なところ、余計な作り込みをしていないことを追加チェックする。

３－４　データと組み合わせ

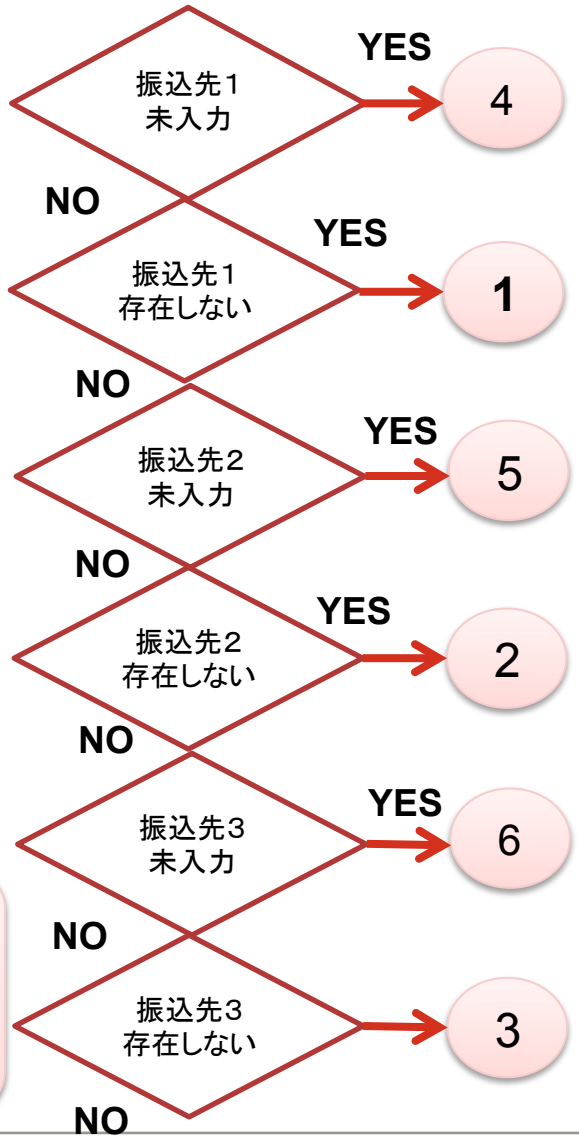
要因		A	B	C
		振込先１情報	振込先２情報	振込先３情報
因子	1	未入力	未入力	未入力
	2	正常	正常	正常
	3	エラー	エラー	エラー

組合せ条件（要因/因子の番号をカンマで区切って1条件/1行で記入。 記入例：要因Aと要因B	
1	A-1/3
2	A-2,B-1/3
3	A-2,B-2,C-1/3

要因		A	B	C	D
		振込先１情報	振込先２情報	振込先３情報	
因子	1	未入力	未入力	未入力	
	2	正常	正常	正常	
	3	エラー	エラー	エラー	

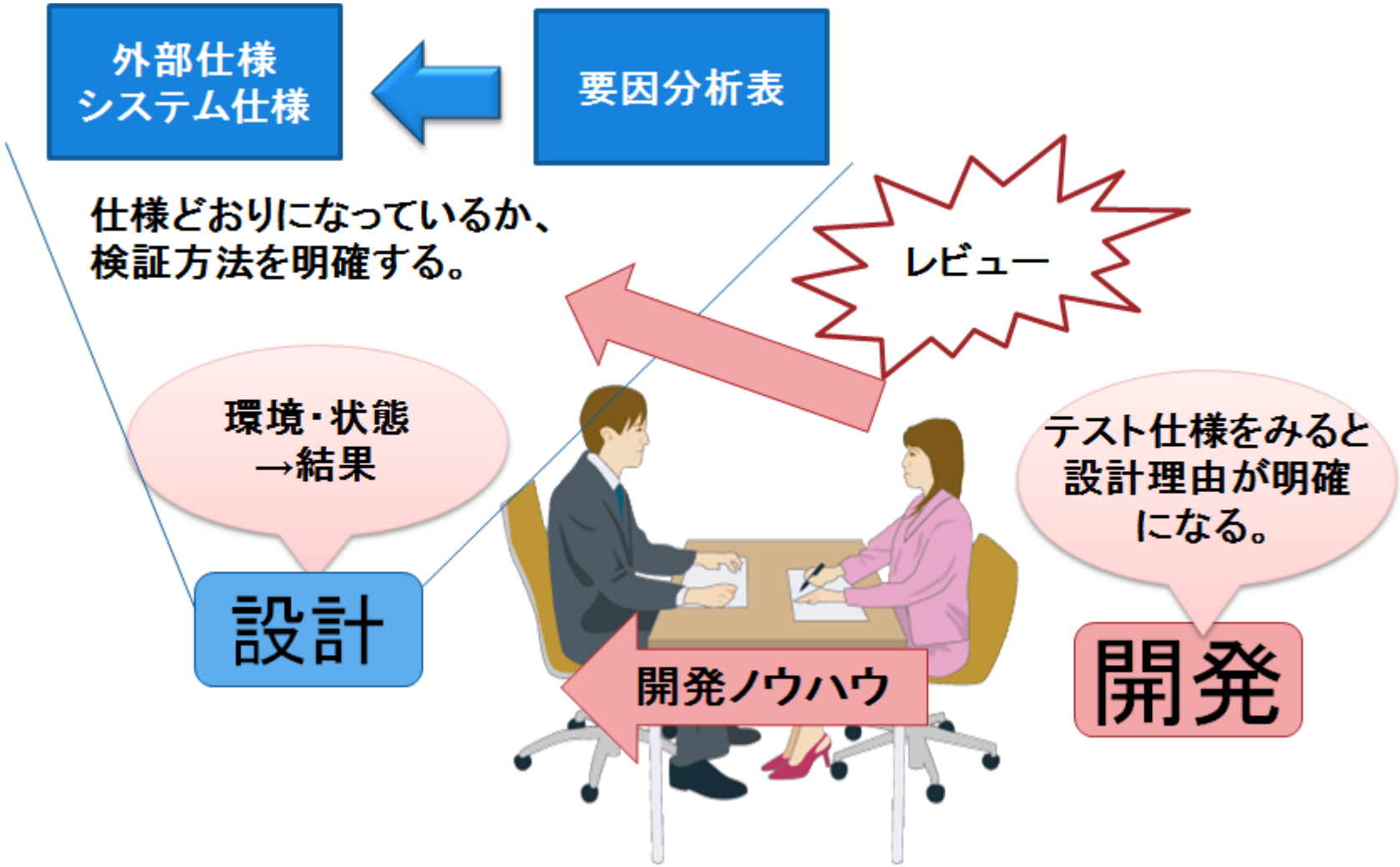
組合せ条件（要因/因子の番号をカンマで区切って1条件/1行で記入。 記入例：要因Aと要因B	
1	A-1/3
2	A-2,B-1/3
3	A-2,B-2,C-1/3
4	A-1,B-2,C-3
5	A-3,B-1,C-2

チェック内容の順序性、
項目の順序性を確認
するテスト追加



３－５　なにが起きているのか

FUJITSU



ソフトウェアテストシンポジウム2014
セッションD2-1　事例発表

FUJITSU
shaping tomorrow with you

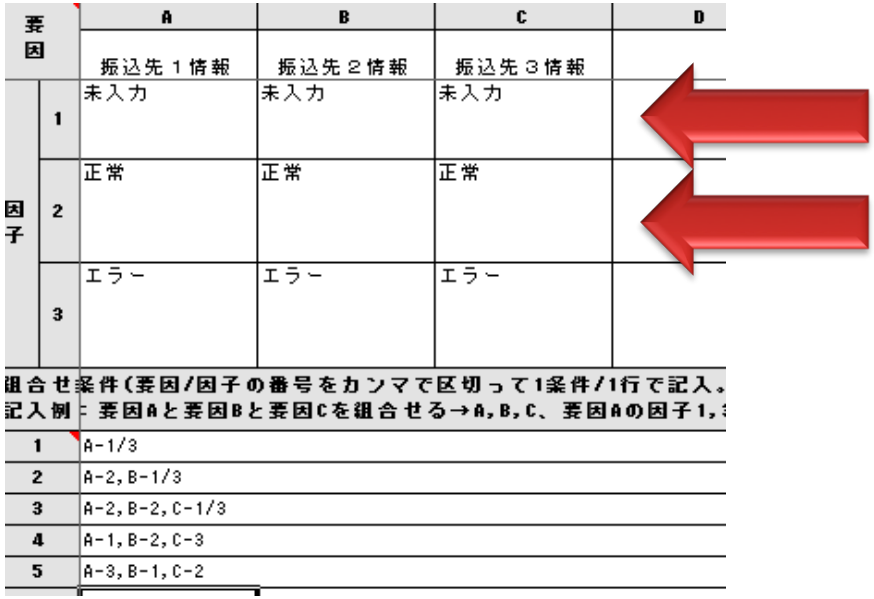
4. 改善策2　パターンを絞り込む

４－１　パターンを絞り込む
４－２　効果
４－３　データと組み合わせ
４－４　何が起きているのか

４－１　パターンを絞り込む

FUJITSU

- 同一のチェック処理をしない
  - A-1、B-3及びC-1のパターンを削る
    - 同一チェックは構造化させる。
    - それでも全パターンできるように考えさせる。



要因		A	B	C	D
		振込先１情報	振込先２情報	振込先３情報	
因子	1	未入力	未入力	未入力	
	2	正常	正常	正常	
	3	エラー	エラー	エラー	

組合せ条件記入例	条件(要因/因子の番号をカンマで区切って1条件/1行で記入。 ：要因Aと要因Bと要因Cを組合せる→A,B,C、要因Aの因子1,
1	A-1/3
2	A-2,B-1/3
3	A-2,B-2,C-1/3
4	A-1,B-2,C-3
5	A-3,B-1,C-2

４－２　効果

FUJITSU

- テスト仕様を決めることによって、プログラムの構造を見直す。
- 組み合わせパターンが最小になるように設計する。
- 危険なところ、余計な作り込みをしていないことを追加チェックする。

４－３　データと組み合わせ

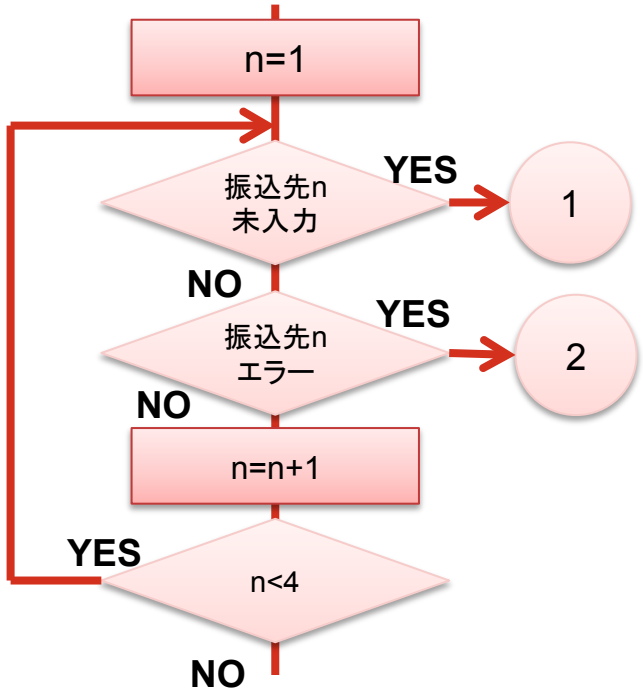
FUJITSU

■ 更に組み合わせを減らす

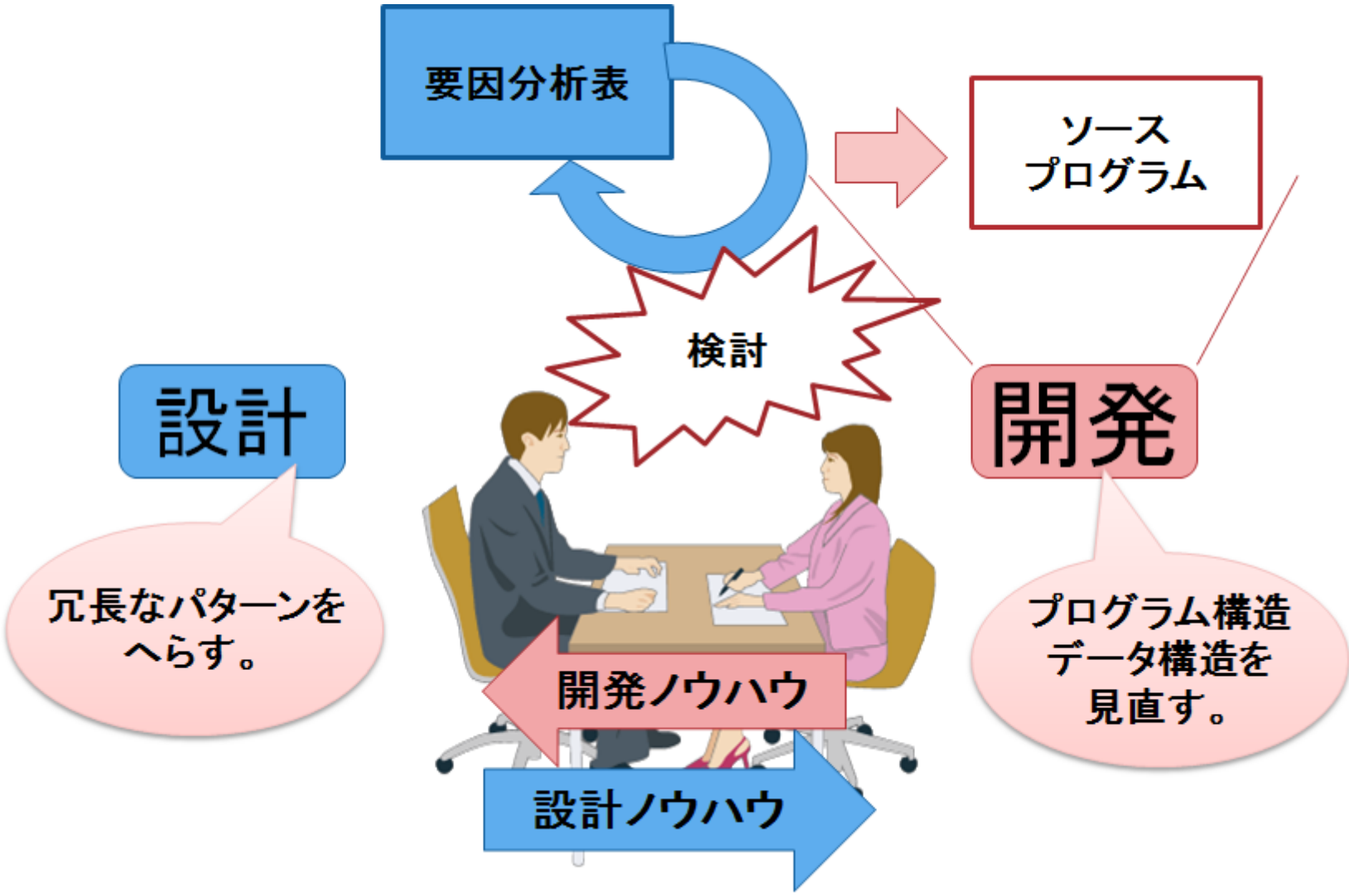
要因		A	B	C
		振込先１情報	振込先２情報	振込先３情報
因子	1	未入力	未入力	未入力
	2	正常	正常	正常
	3	エラー	エラー	エラー

組合せ条件（要因/因子の番号をカンマで区切って1条件/1
記入例：要因Aと要因Bと要因Cを組合せる→A,B,C、要因
1	A-3
2	A-2,B-1
3	A-2,B-2,C-3
4	A-1,B-2,C-3
5	A-3,B-1,C-2

冗長なテストを含めて
テストパターンは
５パターンになる。



４－４　なにが起きているのか



ソフトウェアテストシンポジウム2014
セッションD2-1　事例発表

FUJITSU
shaping tomorrow with you

5. 繰返開発

５－１　現金併用を追加
５－２　追加を繰り返すと
５－３　網羅パターンをつくる
５－４　パターンを絞り込む
５－５　メリット

５－１　現金併用を追加

FUJITSU

➢ 現金併用が指定されたら未入力かチェックする。

■ 現金併用区分

併用なし
振込先1現金
振込先2現金

■ 振込先情報

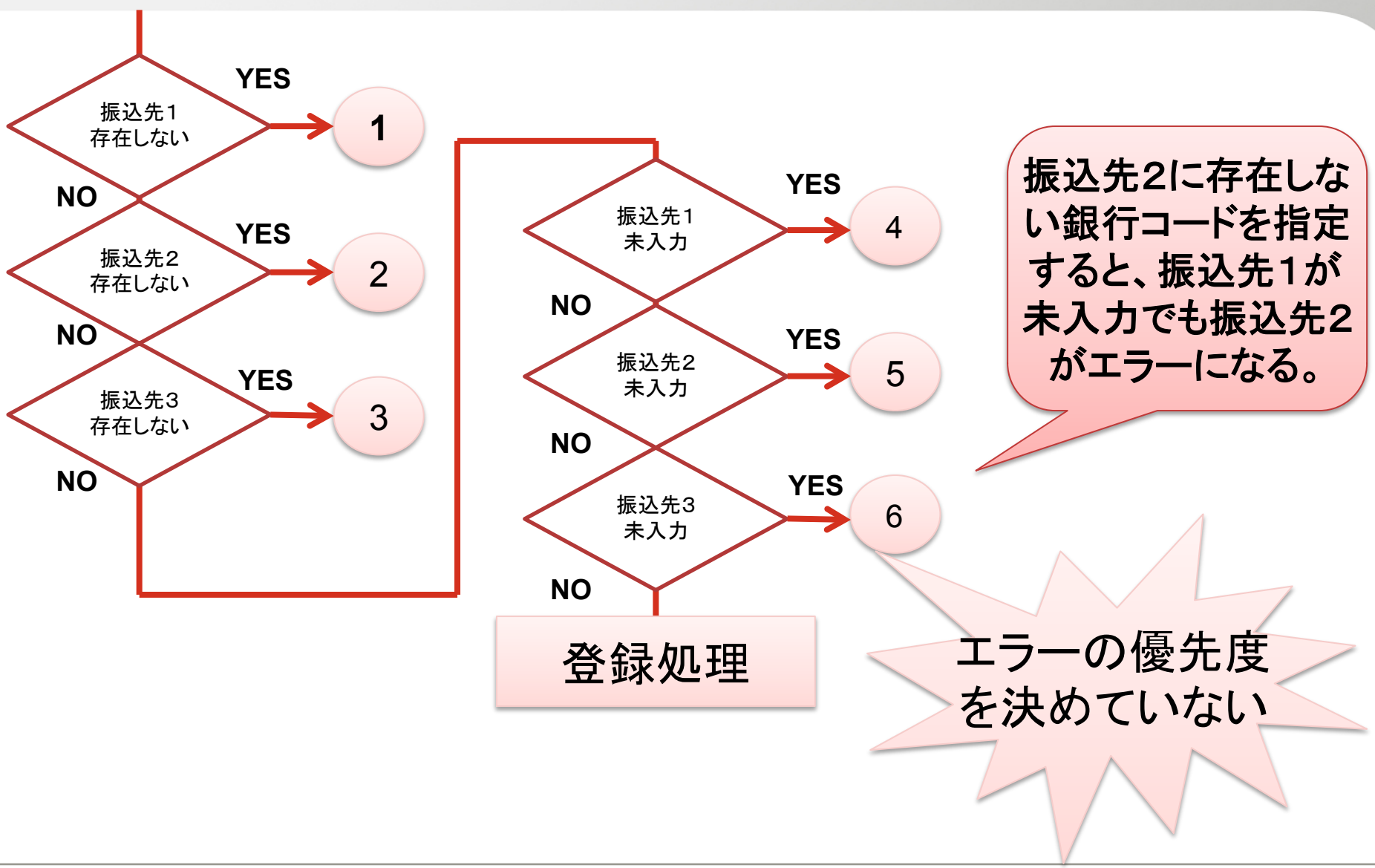
	振込先1		振込先2		振込先3	
銀行コード	0010	AAA銀行	0020	BBB銀行	0030	CCC銀行
店舗コード	111	東京支店	222	神田支店	333	秋葉原店
口座種別	普通預金		普通預金		普通預金	
口座番号	00000001		00000002		0000003	

■ 振込口座名義人

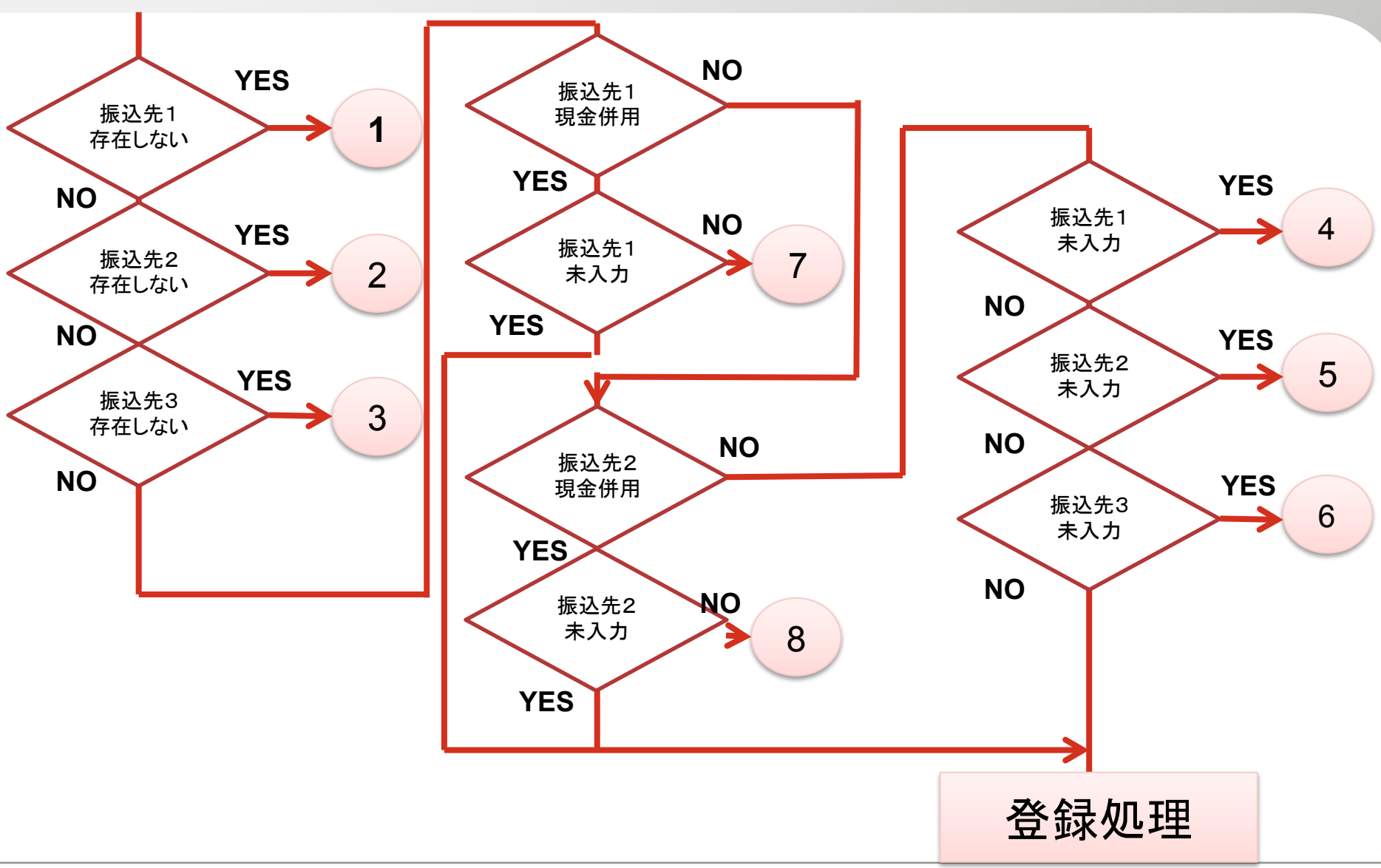
名義人	○○　○○
名義人(カナ)	マルマル　マルマル

登録

５－２　追加を繰り返す１



５－２　追加を繰り返す２



５－２　追加を繰り返す３

FUJITSU

- プログラム構造が複雑になる。
- 全パターンテストするのにコストがかかる。
- ソースコードもテストコードも、リファクタリングしなくてはならない。

５－３　網羅パターンを作る１

FUJITSU

- 網羅できるパターンを作る
  - A-1,B-1/2/3,C-1/2/3,D-1/2/3で設計しない
  - A-1,B-1/3で、振込先１のチェックが最優先
  - A-1,B-2,C-1/3で、振込先１→振込先２→振込先３の順
- テスト仕様を決めることによって、設計の不十分な点を補完する。
- 組み合わせパターンを減らすように設計する。
- 危険なところ、余計な作り込みをしていないことを追加チェックする。

５－３　網羅パターンを作る2

FUJITSU

要因		A	B	C	D
		現金併用区分	振込先１情報 銀行コード	振込先２情報 銀行コード	振込先３情報 銀行コード
因子	1	併用なし	NULL	NULL	NULL
	2	振込先１現金	0010　AAA銀行	0020　BBB銀行	0030　CCC銀行
	3	振込先２現金	9988	9988	9988

組合せ条件（要因/因子の番号をカンマで区切って1条件/1行で記入。複数
記入例：要因Aと要因Bと要因Cを組合せる→A,B,C、要因Aの因子1,3と要因

1	A-1,B-3
2	A-1,B-2,C-1
3	A-1,B-2,C-2,D-3
4	A-1,B-1,C-2,D-3
5	A-1,B-3,C-1,D-2
6	A-2,B-2/3
7	A-2,B-1,C-1/3
8	A-2,B-1,C-2,D-1/3
9	A-2,B-2,C-2,D-3
10	A-2,B-3,C-1,D-2
11	A-3,B-1/3
12	A-3,B-2,C-2/3
13	A-3,B-2,C-1,D-1/3
14	A-3,B-2,C-2,D-3
15	A-3,B-3,C-3,D-2

現金併用区分がない時のパターン

現金併用区分が指定された時のパターン

追加分の順序性確認テスト

５－４　パターンを絞り込む１

FUJITSU

- 同一のチェック処理をしない
  - B-3とC-1のパターンを削る
    - 同一チェックは構造化させる。
    - それでも全パターンできるように考えさせる。
- テスト仕様を決めることによって、プログラムの構造を見直す。
- 組み合わせパターンが最小になるように設計する。
- 危険なところ、余計な作り込みをしていないことを追加チェックする。

５－４　パターンを絞り込む２

FUJITSU

要因		A	B	C	D
		現金併用区分	振込先１情報 銀行コード	振込先２情報 銀行コード	振込先３情報 銀行コード
因子	1	併用なし	NULL	NULL	NULL
	2	振込先１現金	0010　AAA銀行	0020　BBB銀行	0030　CCC銀行
	3	振込先２現金	9988	9988	9988

組合せ条件(要因/因子の番号をカンマで区切って1条件/1行で記入。複数
記入例：要因Aと要因Bと要因Cを組合せる→A,B,C、要因Aの因子1,3と要因

1	A-1,B-3
2	A-1,B-2,C-1
3	A-1,B-2,C-2,D-3
4	A-1,B-1,C-2,D-3
5	A-1,B-3,C-1,D-2
6	A-2,B-2/3
7	A-2,B-1,C-1/3
8	A-2,B-1,C-2,D-1/3
9	A-2,B-2,C-2,D-3
10	A-2,B-3,C-1,D-2
11	A-3,B-1/3
12	A-3,B-2,C-2/3
13	A-3,B-2,C-1,D-1/3
14	A-3,B-2,C-2,D-3
15	A-3,B-3,C-3,D-2

現金併用区分がない時のパターン

現金併用区分が指定された時のパターン

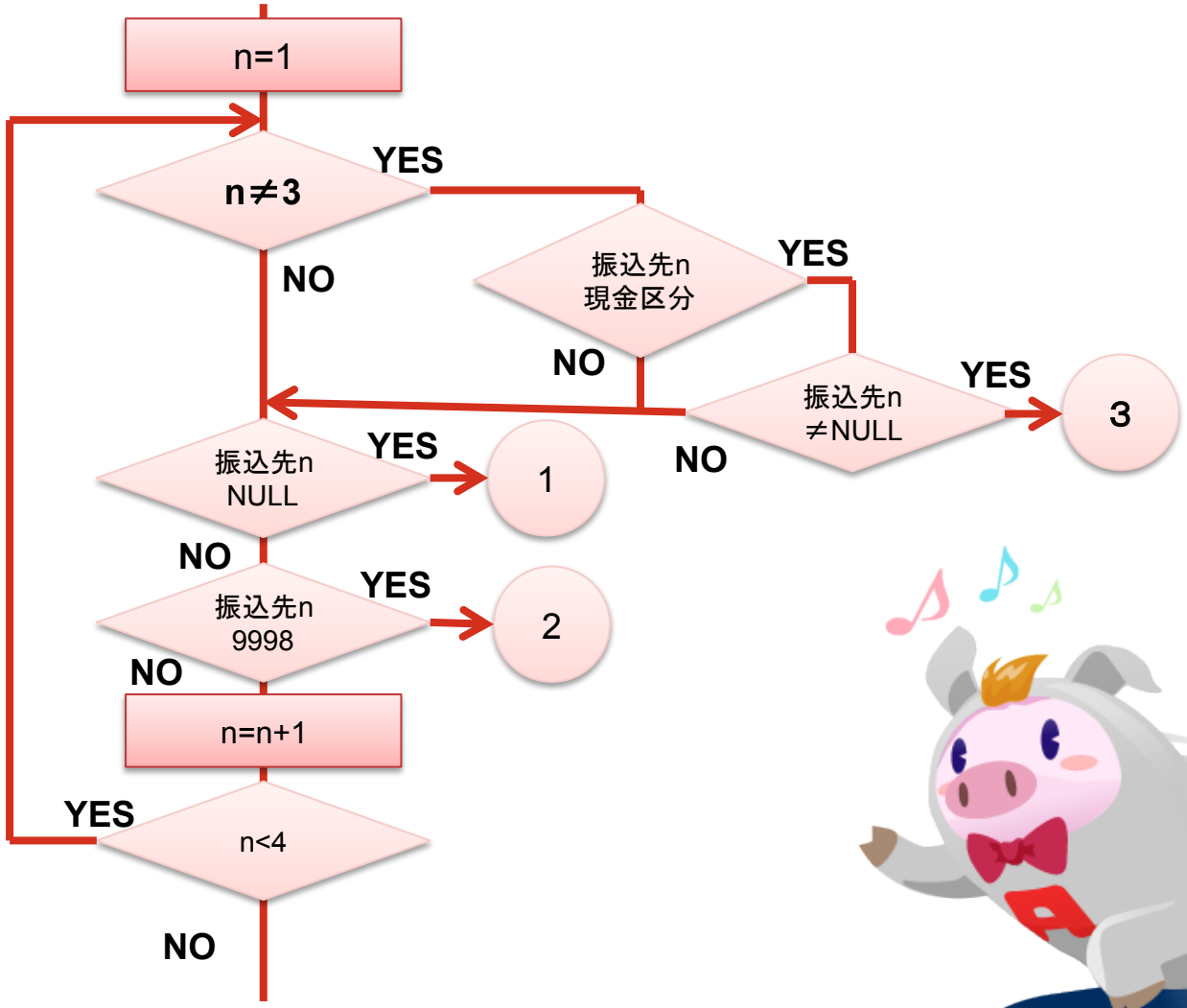
追加分の順序性確認テスト

要因		A	B	C	D
		現金併用区分	振込先１情報 銀行コード	振込先２情報 銀行コード	振込先３情報 銀行コード
因子	1	併用なし	NULL	NULL	NULL
	2	振込先１現金	0010　AAA銀行	0020　BBB銀行	0030　CCC銀行
	3	振込先２現金	9988	9988	9988

組合せ条件(要因/因子の番号をカンマで区切って1条件/1行で記入。複数
記入例：要因Aと要因Bと要因Cを組合せる→A,B,C、要因Aの因子1,3と要因

1	A-1,B-3
2	A-1,B-2,C-1
3	A-1,B-2,C-2,D-3
4	A-1,B-1,C-2,D-3
5	A-1,B-3,C-1,D-2
6	A-2,B-2
7	A-2,B-1,C-3
8	A-2,B-1,C-2,D-1
9	A-2,B-2,C-2,D-3
10	A-2,B-3,C-1,D-2
11	A-3,B-1
12	A-3,B-2,C-2
13	A-3,B-2,C-1,D-3
14	A-3,B-2,C-2,D-3
15	A-3,B-3,C-3,D-2

５－４　パターンを絞り込む



５－５　メリット

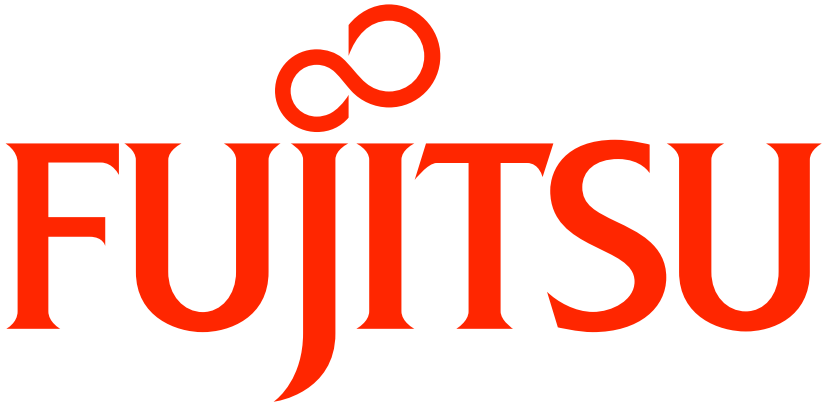
FUJITSU

- テスト技法を設計技術にしていることで、技術者の垣根をなくす。
- ダメなのを作る機会が減る。リファクタリングを考える。
- ソースコード品質を向上させ、プロジェクトの生産性向上及び品質向上を実現する。

注意事項

FUJITSU

- GLOVIAは日本およびその他の国における富士通株式会社の登録商標または商標です。
- Microsoft、Windows、Windows Vistaは米国Microsoft Corporationの米国およびその他の国における登録商標または商標です。
- Word、Excel、Internet Explorerは米国Microsoft Corporationの製品です。
- その他、会社名、製品名、名称等の固有名詞は各社の登録商標または商標です。
- 本資料に記載されているシステム名、製品名称等には、必ずしも商標表示を付記していません。
- 本資料の著作権は富士通マーケティングにあります。



shaping tomorrow with you