「超」やさしいレビュー
～開発とQAに壁はない！～

株式会社ワークスアプリケーションズ
風間 裕也

会社紹介

商　　号	株式会社ワークスアプリケーションズ
設　　立	1996年7月
事 業 概 要	大手企業向け基幹業務パッケージ「COMPANY」および「HUE」の開発・販売・サポート
従 業 員 数	連結 7,599名 (2017年6月末時点)

大手企業向けERPパッケージ市場
7年連続シェアNo.1※
1,300企業グループ超が採用

グローバルR&Dを推進

世界初の人工知能型ビジネスアプリケーション

人工知能によるビッグデータ解析・学習

分散処理技術による高速処理

※市場占有率推移(パッケージ市場) 販売社数シェア
出典:株式会社富士キメラ総研 ソフトウェアビジネス新市場 2011-2017年版(2010-2016年度)

Agenda

- 今回の前提共有のためのアンケート
- レビューを行う目的
- 実際にレビュアーを体験してみよう
- レビューに参加するハードルを下げるコツ
- 今後のレビューのかたち

本日のターゲットとゴール

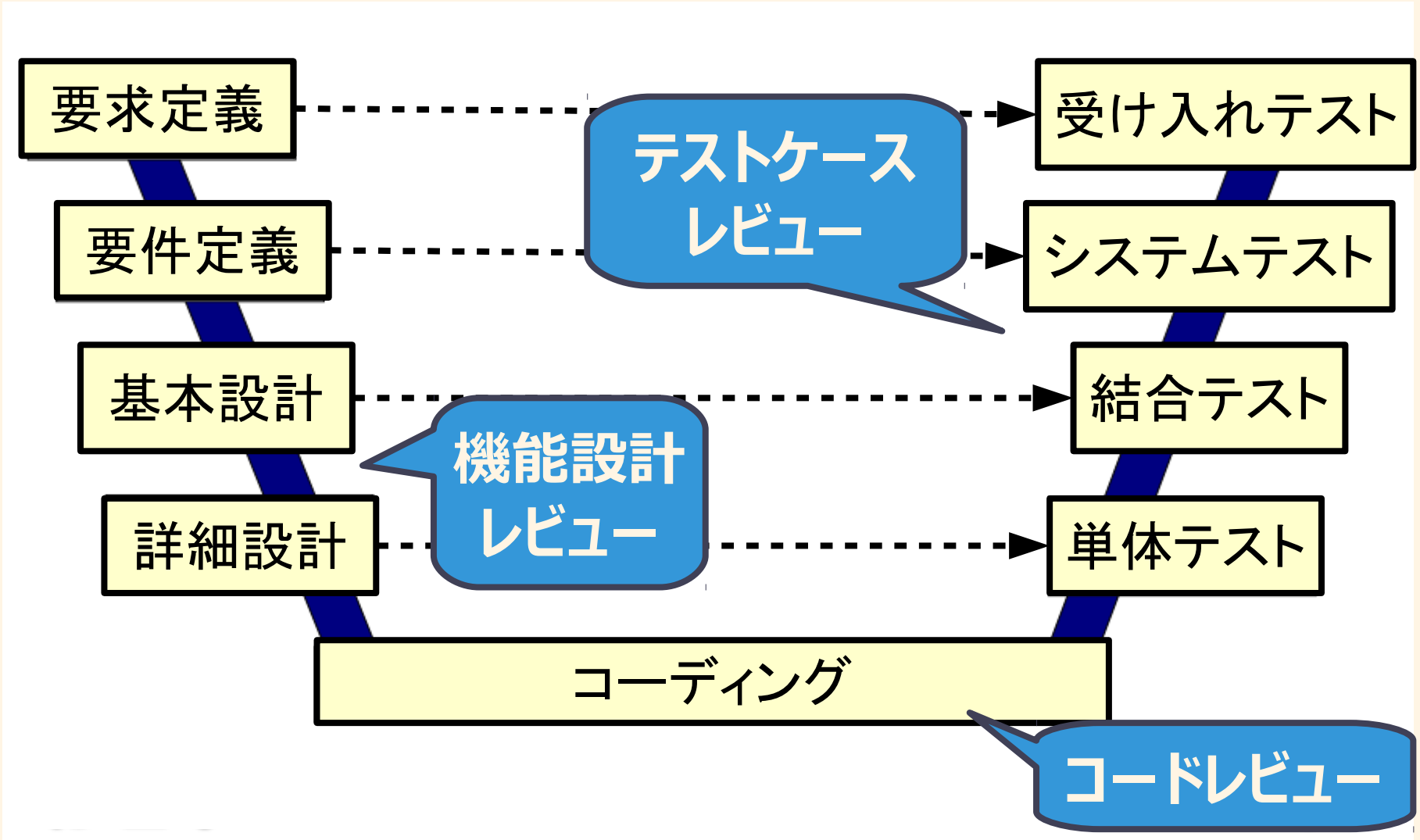
- 発表を聞いてほしいターゲット
  - とにかくレビューをこれからしないといけない
  - レビューをやっているが効果が薄いと悩んでいる
- 発表のゴールとして感じてほしいこと
  - 気軽に行うレビューの方法もある
  - 易しい レビューを難しく考える必要はない
  - 優しい レビュイーに対して厳しくあたる必要はない

今回の前提共有のためのアンケート

「レビュー」といっても様々

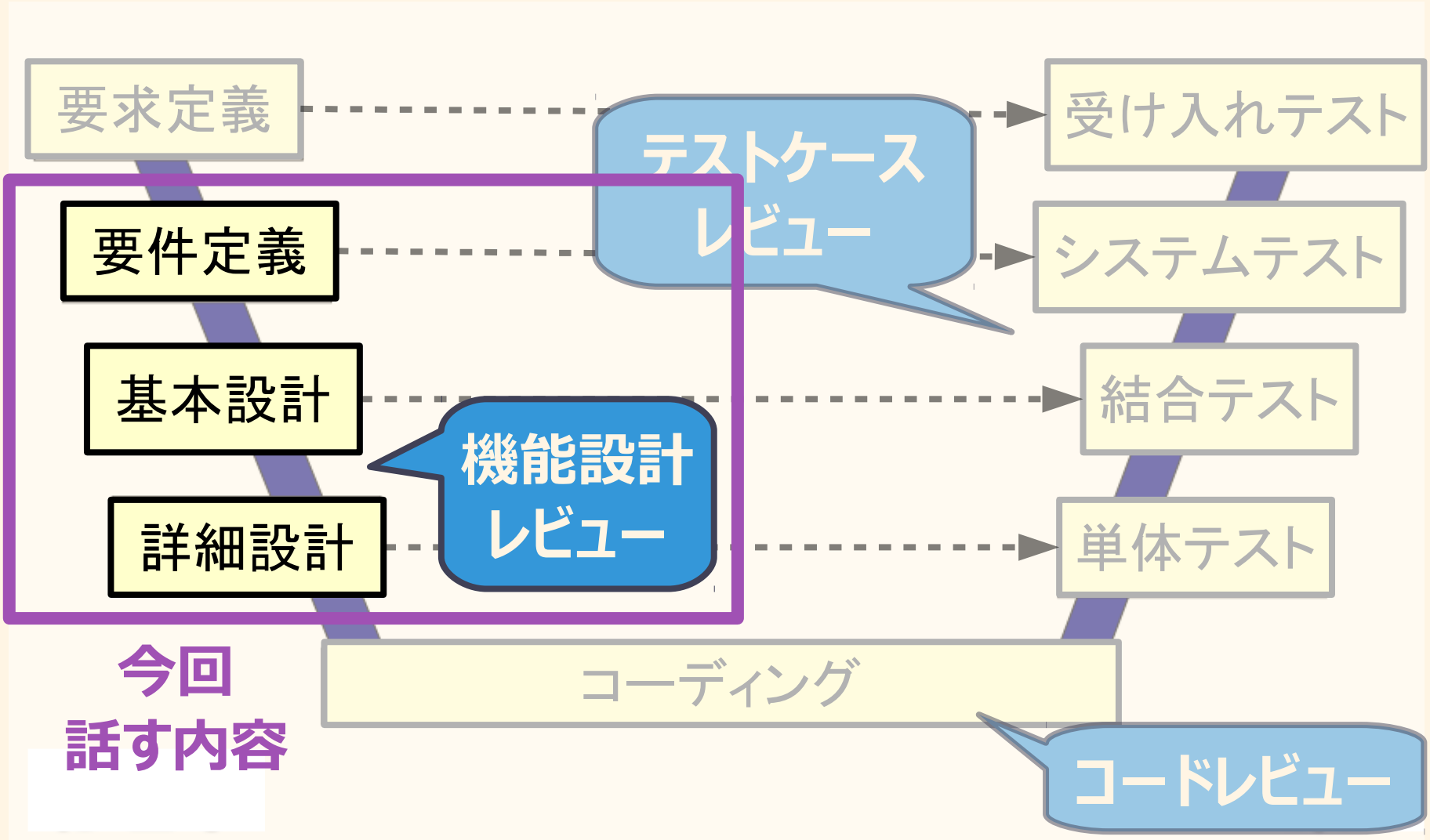
- 「レビュー」は、各組織で様々な形、方法があります
- この章では、皆さんの組織での
  レビュー状況についてアンケートを取ります
  - レビューの種類
  - レビュー方法
  - リーディング技法
  - レビューを行うタイミング
- 複数当てはまる、少しでも当てはまる場合は挙手を！
- これにより**前提の認識合わせ**を行います

質問1：レビューの種類



出典：JaSST'12 Tokyo 鈴木三紀夫「Wモデルとは何か」
http://jasst.jp/symposium/jasst12tokyo/pdf/D4-1.pdf を元に作成

質問1：レビューの種類



出典：JaSST'12 Tokyo 鈴木三紀夫「Wモデルとは何か」
http://jasst.jp/symposium/jasst12tokyo/pdf/D4-1.pdf を元に作成

質問2：レビュー方法

- インスペクション…最も**公式的**なレビュー
- チームレビュー…**チーム**により実施される
- パスアラウンド…成果物をレビュアーへ**配布、回覧**する
- ラウンドロビンレビュー…全員が**順番に**司会者とレビュアーになる
- ピアデスクチェック…**熟練者**のレビュアーと実施する
- ウォークスルー…**形式的ではなく**、参加者が質問やコメントする
- ピアレビュー…作成者以外と**気軽に行う**
- アドホックレビュー…近くの**同僚に声をかける**非公式なレビュー
- ペアプログラミング…**プログラミング時**に他者視点を取り入れる

出典：SQuBOK策定部会（2014）『ソフトウェア品質知識体系ガイド（第2版） －SQuBOK® Guide V2－』オーム社.

質問2：レビュー方法

- インスペクション…最も公式的なレビュー
- チームレビュー…チームにより実施される
- パスアラウンド…成果物をレビュアーへ配布、回覧する
- ラウンドロビンレビュー…全員が順番に司会者とレビュアーになる
- ピアデスクチェック…熟練者のレビュアーと実施する **今回話す内容**
- ウォークスルー…**形式的ではなく**、参加者が質問やコメントする
- ピアレビュー…作成者以外と**気軽に行う**
- アドホックレビュー…近くの**同僚に声をかける**非公式なレビュー
- ペアプログラミング…プログラミング時に他者視点を取り入れる

出典：SQuBOK策定部会（2014）『ソフトウェア品質知識体系ガイド（第2版） －SQuBOK® Guide V2－』オーム社.

質問3：リーディング技法

- CBR(Checklist-Based Reading)
  - **チェックリスト**を利用して読む技法
- DBR(Defect-Based Reading)
  - 固有の**欠陥タイプを摘出**することに集中して読む技法
- UBR(Usage-Based Reading)
  - 要求レベルの**ユースケースを使用**して読む技法
- PBR(Perspective-Based Reading)
  - 「設計者」「テスター」「顧客」の**視点を割り当てて**読む技法
- アドホック

出典：SQuBOK Review 2017　SQuBOK V3 研究チーム「レビュー技術動向」
https://www.juse.or.jp/upload/files/SQuBOK_Review2017_2_20171026.pdf#page=6

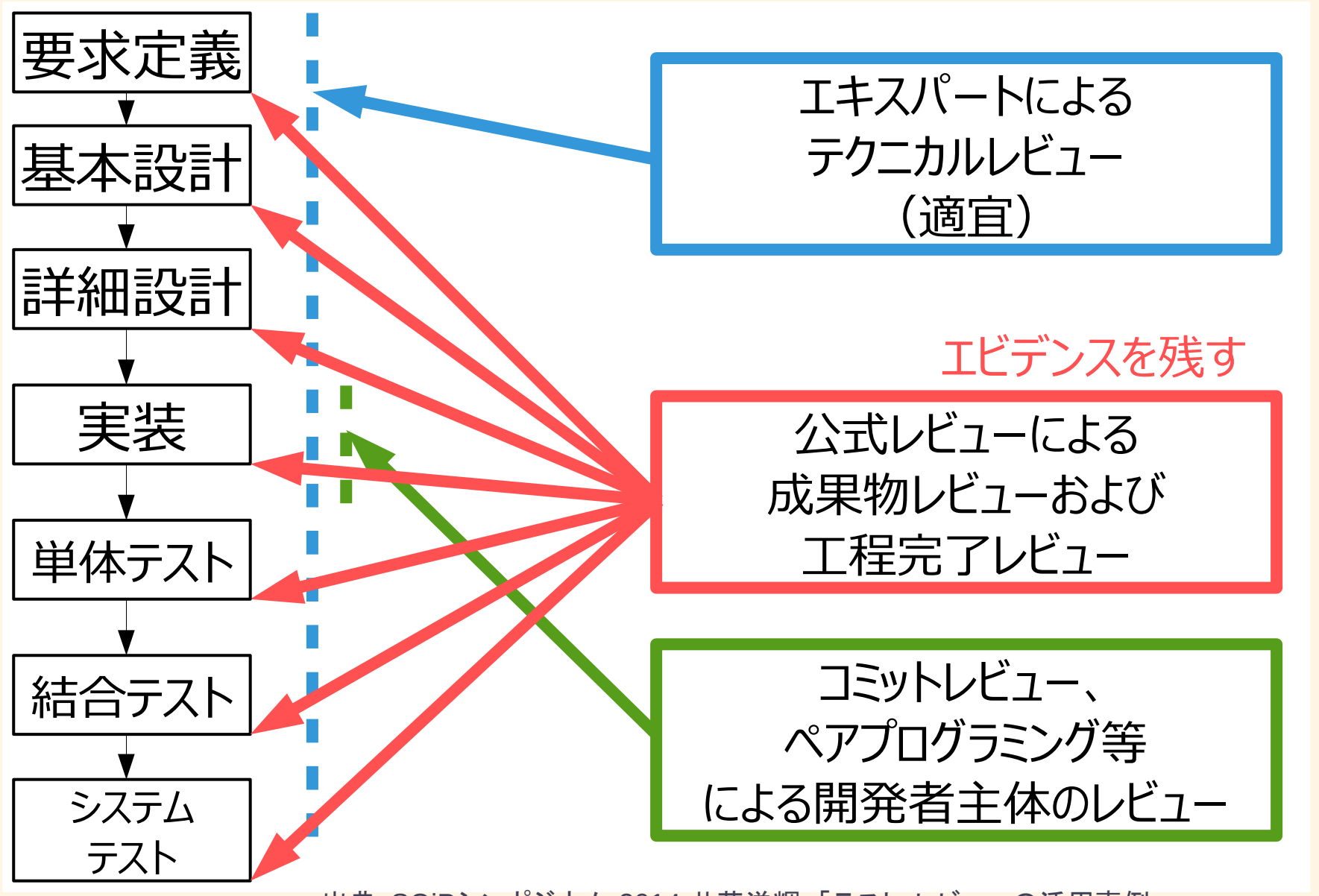
質問3：リーディング技法

- CBR(Checklist-Based Reading)
  - チェックリストを利用して読む技法
- DBR(Defect-Based Reading)
  - 固有の欠陥タイプを摘出することに集中して読む技法
- UBR(Usage-Based Reading)
  - 要求レベルのユースケースを使用して読む技法
- PBR(Perspective-Based Reading)
  - 「設計者」「テスター」「顧客」の**視点を割り当てて**読む技法
- アドホック

今回話す内容

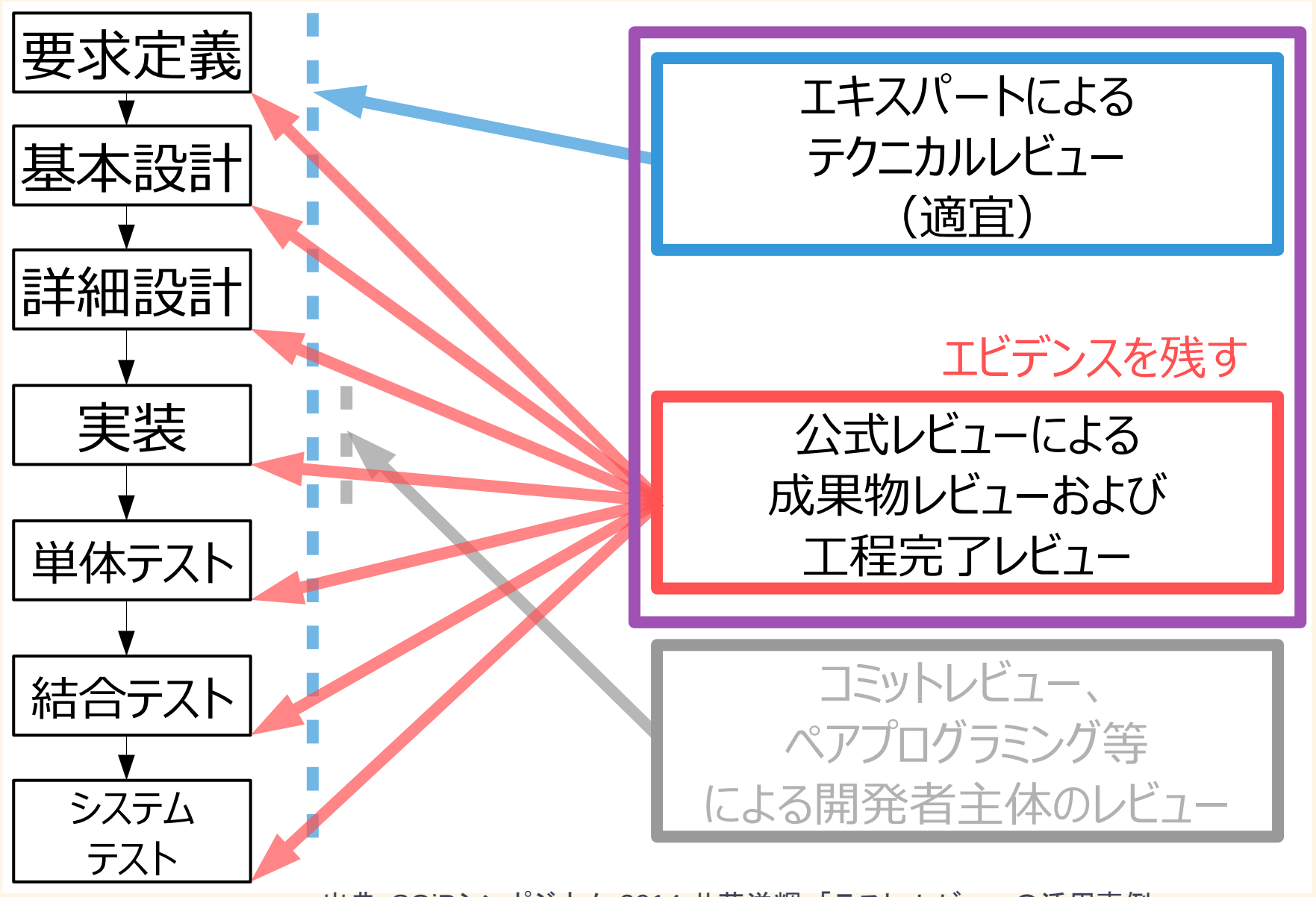
出典：SQuBOK Review 2017　SQuBOK V3 研究チーム「レビュー技術動向」
https://www.juse.or.jp/upload/files/SQuBOK_Review2017_2_20171026.pdf#page=6

質問4：レビューを行うタイミング



出典：SQiPシンポジウム 2014 井芹洋輝 「テスト・レビューの活用事例」
https://www.slideshare.net/goyoki/sqip2014/4 を元に作成

質問4：レビューを行うタイミング



今回話す内容

出典：SQiPシンポジウム 2014 井芹洋輝 「テスト・レビューの活用事例」
https://www.slideshare.net/goyoki/sqip2014/4 を元に作成

「レビュー」といっても様々

- 「レビュー」は、各組織で様々な形、方法がある
  - レビューの種類
  - レビュー方法
  - リーディング技法
  - レビューを行うタイミング
- 今回お話しするレビューの特徴は以下の通り
  - **機能設計**レビュー
  - **気軽に**レビューする
  - **作る途中**でもレビューする

レビューを行う目的

SQuBOK Guide
（ソフトウェア品質知識体系ガイド）

レビューとは一般的に、ソフトウェア開発工程全般で
行われる**見直し作業であり、**
関係者が参加し多角的に検討することで、
論理の客観性と透明性、構造の妥当性、
フィールドへの適応性等を評価/確認する。

第1版

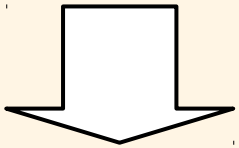
出典：SQuBOK策定部会(2007)
『ソフトウェア品質知識体系ガイド ―SQuBOK® Guide―』オーム社.

SQuBOK Guide
（ソフトウェア品質知識体系ガイド）

レビューとは一般的に、ソフトウェア開発工程全般で
行われる**見直し作業であり、**
関係者が参加し多角的に検討することで、
論理の客観性と透明性、構造の妥当性、
フィールドへの適応性等を評価/確認する。

第1版

出典：SQuBOK策定部会(2007)
『ソフトウェア品質知識体系ガイド ―SQuBOK® Guide―』オーム社.



レビューとは一般的に、ソフトウェア開発工程全般で
行われる**評価及び確認の作業のことであり、**
関係者が参加し多角的に検討することで、
論理の客観性と透明性、構造の妥当性、
フィールドへの適応性などを評価し、確認する。

第2版

出典：SQuBOK策定部会（2014）
『ソフトウェア品質知識体系ガイド（第2版） －SQuBOK® Guide V2－』オーム社.

SQuBOK Guide
（ソフトウェア品質知識体系ガイド）

レビューとは一般的に、ソフトウェア開発工程全般で
行われる**見直し作業であり、**
関係者が参加し多角的に検討することで、

第1版

レビューは見直すだけではなく、
次へ繋げる作業

行われる**評価及び確認の作業のことであり、**
関係者が参加し多角的に検討することで、
論理の客観性と透明性、構造の妥当性、
フィールドへの適応性などを評価し、確認する。

第2版

出典：SQuBOK策定部会（2014）
『ソフトウェア品質知識体系ガイド（第2版） －SQuBOK® Guide V2－』オーム社.

QAが行うべきこととは

QAとは（『ベタープログラマ』より）

彼らの名前は、理由があって「テスト部門」ではなく「QA（品質保証）」なのです。彼らの役割は、ロボットのようにボタンを押すことではありません。**品質を製品に作り込むこと**なのです。

そのために、**QAは開発プロセスの最後ではなく、プロセス全体を通して深く関与していなければなりません。**

- 作られるものを理解して形づくるために、ソフトウェアの仕様書に関与します。
- 作られるものをテスト可能にするために設計と開発に貢献します。
- 当然ですが、テストフェーズでは深く関与します。
- そして、最終の物理的なリリースにも深く関与します。つまり、テストされたものが実際にリリースされて配置されるようにします。

出典：JaSST'18 Tokyo 柴田芳樹「私が経験したソフトウェアテストの変遷」
http://jasst.jp/symposium/jasst18tokyo/pdf/A7.pdf#page=45

レビューに期待していること

- **製品に品質を作り込みたい**
  - バグを早めに**防ぎたい**
  - コストのかからない**改善活動**をしたい
  - レビューの後工程にある**テストの材料**を得たい
- **進捗度合い(ヤバさ)を知りたい**
- **製品や開発を新人に理解してもらいたい**
- **製品を良くしたい**

レビューって難しい？

- ここまで、レビューの目的、期待していることを示しました
- それでもまだ以下のように思っている人はいるのでは？
  - レビューのイメージがまだよく分からない 難しい？
  - レビュアーになるのが難しそう 難しい？
  - QAがレビューに参加しても意味がなさそう 難しい？

- **次の章では、レビューをどのように行うか**
  実際に体験してみましょう

実際にレビュアーを体験してみよう

レビュー体験の目的

易しい

- レビュー参加の**ハードルは低い**と実感してもらう
- 自分でも**レビュアーとして指摘できる**と感じてもらう
- レビューに**参加する価値**を感じてもらう

体験してもらう前に、
まずは**レビュアーとしての心構え**をお伝えします。

易しい

「レビュアー」という役割に気負わない

- 自分の知らないアクターになって臨む必要はない
- QAであればQAとして、開発者であれば開発者としてレビューに参加すればいい
- **気軽な気持ちで参加しよう**
  - 難しく考えないこと！

優しい

「レビュイー」に完璧を求めない

- **最初から完璧な仕様書を求めなくていい**
  - いつまでもレビューが開始されなくなる
  - 完璧に近づける作業がレビュー
- **書きたいものから書けばいい**
  - 最初の一歩を踏み出そう（勢いも大事）
- **書いて迷ったときに聞けばいい**
  - 後になるほど、完璧なものを求められる謎のプレッシャー…
  - 途中経過をレビューしてもらうのもアリ

例題

次の仕様に対してどんなことが気になりますか。

- パスワードは4文字以上12文字以下の
  英数字のみを許容する

- パスワードを3分以内に4回以上間違って入力すると
  アカウントを5分間ロックする

出典：第7回Ques 山﨑崇「概説テスト分析」
https://www.slideshare.net/takashiyamasaki378/ss-55384920/25 を元に作成

文字列長

パスワードは**4**文字以上**12**文字以下の

英数字のみを許容する

文字種

誤入力期間管理

誤入力回数管理

パスワードを**3**分以内に**4**回以上間違って入力すると

アカウントを**5**分間　ロックする

ロック保持期間　**状態遷移**

文字列長

パスワードは**4**文字以上**12**文字以下の

許容しないとどうなる？

(ボタン制御orエラー画面)

英数字のみを許容する

文字種　　**誤入力**　　**誤入力**

期間管理　**回数管理**

パスワードを**3**分以内に**4**回以上間違って入力すると

5回目の入力はどうなる？

アカウントを**5**分間　ロックする

ロック保持期間　**状態遷移**

例題を振り返って…

1. あなたと隣の人で気になる内容が違う
    - 自分が気付いた内容は**伝えましょう**
    - 隣の人が気付かない内容を**補完し合う**ことができます

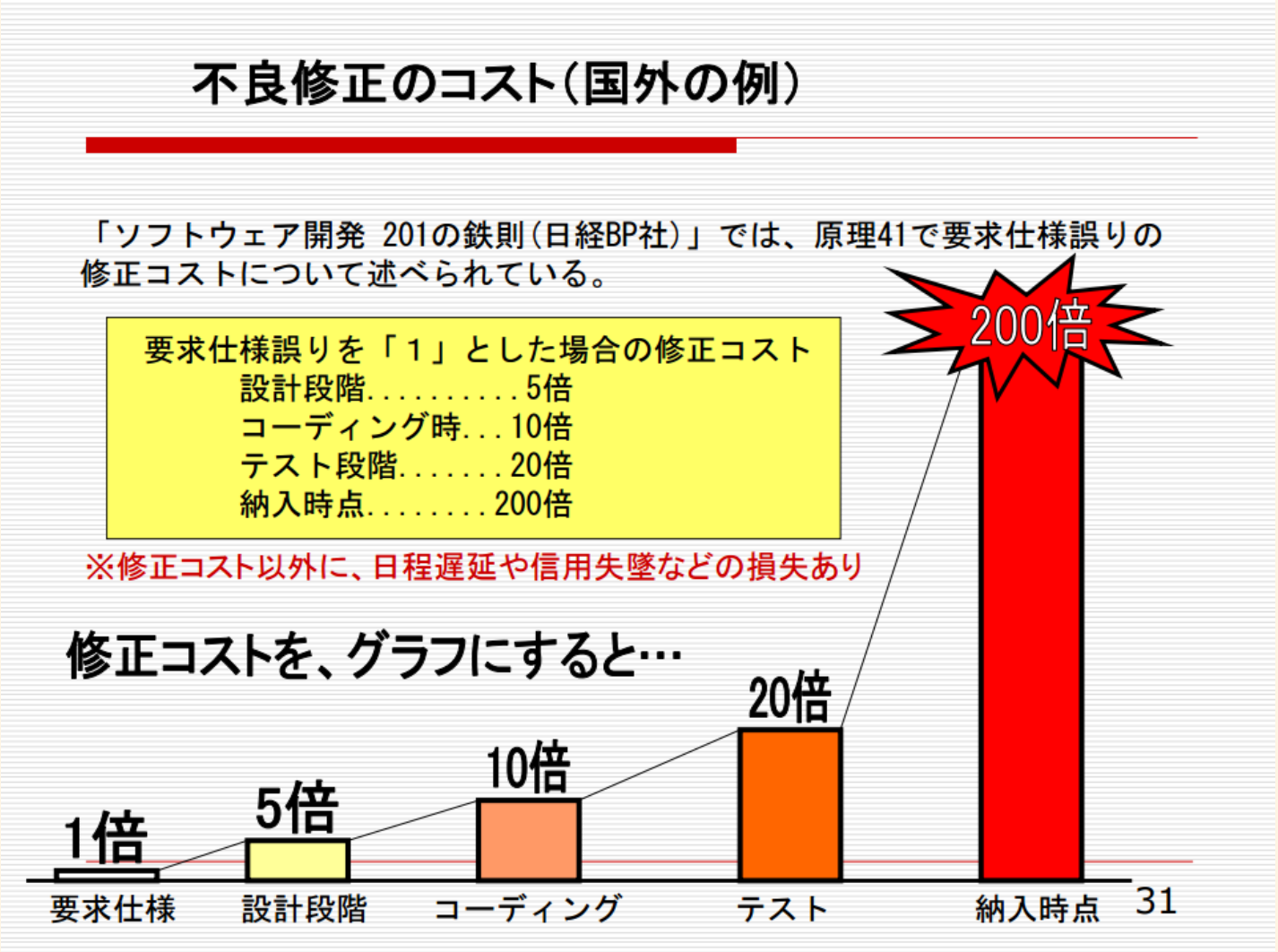
2.

例題を振り返って…

1. あなたと隣の人で気になる内容が違う
   - 自分が気付いた内容は伝えましょう
   - 隣の人が気付かない内容を補完し合うことができます

2. この例では**何もプログラムを書いていません**
   - 不具合を**実装前に防ぐ**ことができる例です
   - もしもこの時点で指摘できれば、
     総コストを削減できます

早めにバグを防ぎたい理由



出典:JASPIC SPIJapan2009 奈良 隆正「ソフトウェア品質保証の方法論、技法、その変遷」
http://www.jaspic.org/event/2009/SPIJapan/keynote/SJ9keynote.pdf#page=31

この章でやったこと

- 簡単な題材で**レビューを体験**してもらった
- 成果物に対して**たくさん疑問点が出た**（はず）
  - 隣の人と違う指摘ができた人
  - 隣の人と同じ指摘しかできなかった人
- 価値があると感じたならば、例題を持ち帰ってください

- 次の章では、社内の人（開発者・QAなど）に**気軽にレビューに参加してもらう**ための話をお伝えします

レビューに参加する
ハードルを下げるコツ

～易しく優しいレビューにする～

優しい

レビューとは相談である

- レビューは相談、確認の場
  - 承認でも保証でも批判でも勝負でもない
- 自分の不足しているところを補うための相談
  - すべて理解してほしいわけではない
- 自分の知識や観点や経験を活用
  - 知らないアクターになって臨む必要はない

優しい

レビューとは相談である

- レビューは相談、確認の場
  - 承認でも保証でも批判でも勝負でもない

レビュアーは1人ではないのでなんとかなる

  - 知らないアクターになって臨む必要はない

易しい

こんな知識や経験を持っていませんか？

- 似たような**過去**の不具合(トラブル)事例
- **お客様**がどんなことに困っているか
- 新人が**よくやらかす**事例
- コードや設計の**アンチパターン**
- その機能の**仕様バグ**や**裏仕様**や**歴史**
- 世の中の**あるある**事例
- ベテランが**いつも気にしている**ポイント

易しい

こんな知識や経験を持っていませんか？

- 似たような**過去**の不具合(トラブル)事例
- お客様がどんなときに困っているか
- 世の中の**あるある**事例
- ベテランが**いつも気にしている**ポイント

何か1つでもあれば
レビュアーになれる

優しい

レビューをすると
みんなの関心事になる

- レビューをする**前**
  - 成果物を知っているのは作成者本人だけ
  - 成果物は**自己満足**(Self satisfaction)にすぎない
- レビューをした**後**
  - みんなが成果物に関心を持ち、**みんなで改善しよう**とする
  - **チーム満足**(Ourselves satisfaction)を目指す

全員が**「もっと製品を良くしたい！」**と思っている

- これを理解していれば、レビュイーを攻撃しませんよね？

レビューを有効活用するために

易しい
- 思ったことは**素直に**伝えよう
  - 知らないから言わないはダメ
  - 「分からない」と言うことも大事
  - 正常に、疑問を持とう

易しい
- QAとしての**不安感**も伝えよう
  - 実装前で気付くテスト観点だったりする
  - 勘違いでも問題ない（不安を解消できる）

優しい
- **製品を良くするために**レビューをしよう
  - 承認でも保証でも批判でも勝負でもない
  - 楽しくやろう！

優しい

実際に開発リーダーから貰ったメール

今回はレビュイーである新人にとって初めての設計レビューでした。

まだ同じチームになって2ヶ月ほどですが、
あんなにも緊張した彼を見たのは初めてでした 笑

レビュー後に「やってみてどうだった？」と聞いたところ、
「いや～楽しかったです。」と返してくれました。

理由は、**色々な視点でフィードバックをもらえたから**だそうです。
彼にとっては非常に価値のある有意義なレビューとなったようです。

優秀なQAは優秀な開発者を育てますし、
優秀な開発者は優秀なQAを育てますよね。

これからもビシバシ気になる点は指摘していただければと思います。

今後のレビューのかたち

ここからは未来の話

https://www.freeimages.com/photo/blank-sign-1-clipping-path-1445187

開発技術は螺旋構造で変化している

変わるもの × 変わらないもの

- 技術の変化の歴史は一見すると **振り子** に見える
- でも実は **螺旋** 構造。同じところには戻ってこない
- **差分** と、それを **可能にした技術** が重要
- 「それは既に20年前に通った道だ」は老害発言かもしれない
- それはそうと、変わらないように見える強固なものもある

出典：Developers Summit2018 和田卓人「技術選定の審美眼」
https://speakerdeck.com/twada/understanding-the-spiral-of-technologies?slide=10

機能設計レビューを取り巻く環境は変化している

https://pixabay.com/ja/ux-プロトタイピング-デザイン-ウェブデザイン-アプリケーション-788002/

アジャイルソフトウェア開発宣言

私たちは、ソフトウェア開発の実践あるいは実践を手助けをする活動を通じて、
よりよい開発方法を見つけだそうとしている。この活動を通して、私たちは以下の価値に至った。

プロセスやツールよりも**個人と対話**を、
包括的なドキュメントよりも**動くソフトウェア**を、
契約交渉よりも**顧客との協調**を、
計画に従うことよりも**変化への対応**を、

価値とする。すなわち、**左記のことがらに価値がある**ことを
認めながらも、私たちは**右記のことがらにより価値をおく**。

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler
James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick
Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

出典：アジャイルソフトウェア開発宣言 http://agilemanifesto.org/iso/ja/manifesto.html

アジャイルソフトウェア開発宣言

私たちは、ソフトウェア開発の実践あるいは実践を手助けをする活動を通じて、
よりよい開発方法を見つけだそうとしている。この活動を通して、私たちは以下の価値に至った。

プロセスやツールよりも**個人と対話**を、

易しい

アジャイルでは、ドキュメントよりも
動くソフトウェアを重視し、
変化への対応を重視している。

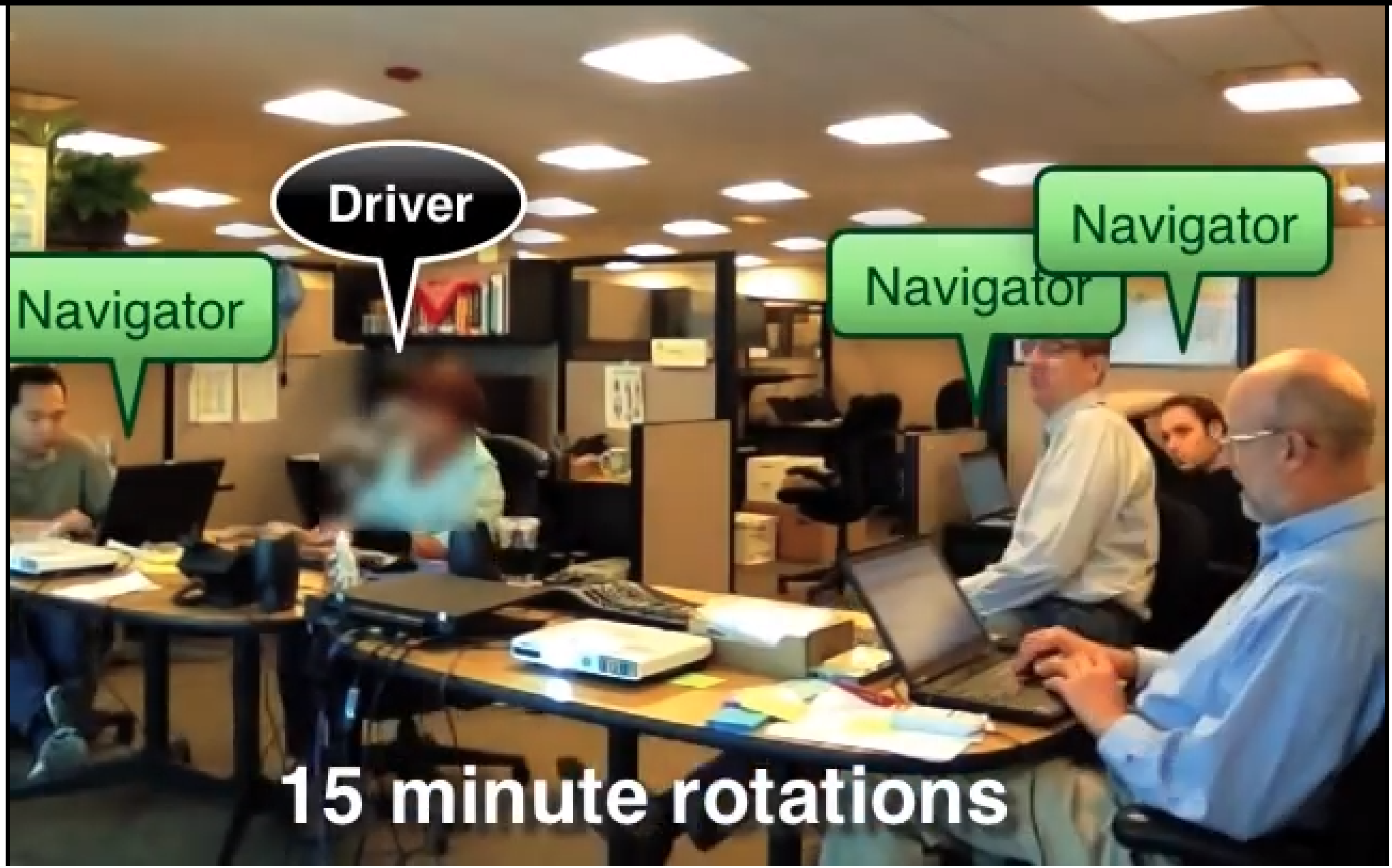
※「ドキュメントが無くていい」とは言っていない。

→それに合わせて、**レビューも気軽なものに変化**

Martin Fowler　　Brian Marick

出典：アジャイルソフトウェア開発宣言 http://agilemanifesto.org/iso/ja/manifesto.html

黒船来襲：モブプログラミング



出典：A day of Mob Programming　https://www.youtube.com/watch?v=p_pvslS4gEI

モブプログラミングは常に同期している

モブワーク

ex.
設計、レビュー、ノウハウ共有

同期していないではなくて
常に同期している

Rakuten

出典：HDIfes#06 開発・デザインの文化をつくる 及部敬雄「我々は文化とどう向き合うべきか-モブプログラミングとチームと文化」
https://speakerdeck.com/takaking22/wo-hawen-hua-todouxiang-kihe-ubekika-mobupuroguramingutotimutowen-hua-number-hdifes?slide=54

モブプログラミングは常に同期している

モブプロを始めてなくなったもの

- ✓ 朝礼
- ✓ コミュニケーションのためのミーティング
- ✓ 合意形成のためのドキュメント
- ✓ レビュー全般（ソースコード、ドキュメント ..）
- ✓ Git Flow
- ✓ カンバン
- ✓ プランニングポーカー（見積もり）

これらはすべて同期作業

Rakuten

出典：HDIfes#06 開発・デザインの文化をつくる 及部敬雄「我々は文化とどう向き合うべきか-モブプログラミングとチームと文化」
https://speakerdeck.com/takaking22/wo-hawen-hua-todouxiang-kihe-ubekika-mobupuroguramingutotimutowen-hua-number-hdifes?slide=56

モブプログラミングは常に同期している

モブプロを始めてなくなったもの

- ✓ 朝礼
- ✓ コミュニケーションのためのミーティング
- ✓ 合意形成のためのドキュメント
- ✓ レビュー全般（ソースコード、ドキュメント ..）
- ✓ Git Flow

易しい

常にチームとして活動することで、
レビューという工程が無くなる！
ただし、レビュー行為（＝相談、指摘）は残っている

Rakuten

出典：HDIfes#06 開発・デザインの文化をつくる 及部敬雄「我々は文化とどう向き合うべきか-モブプログラミングとチームと文化」
https://speakerdeck.com/takaking22/wo-hawen-hua-todouxiang-kihe-ubekika-mobupuroguramingutotimutowen-hua-number-hdifes?slide=56

モブプログラミング中のレビュー行為も螺旋構造の一つ？

- インスペクション…最も公式的なレビュー
- チームレビュー…チームにより実施される
- パスアラウンド…成果物をレビュアーへ配布、回覧する
- ラウンドロビンレビュー…全員が**順番に**司会者とレビュアーになる
- ピアデスクチェック…熟練者のレビュアーと実施する
- ウォークスルー…形式的ではなく、参加者が質問やコメントする
- ピアレビュー…作成者以外と気軽に行う
- アドホックレビュー…近くの同僚に声をかける非公式なレビュー
- ペアプログラミング…プログラミング時に他者の視点を取り入れる

出典：SQuBOK策定部会（2014）『ソフトウェア品質知識体系ガイド（第2版） －SQuBOK® Guide V2－』オーム社.

おわりに

まとめ

- レビューといっても様々なものがある
  - 今回は気軽に行うレビューを紹介した

易しい
- 「レビュアー」という役割に**気負わない**
  - **複数人で補完**し合えば良い

優しい
- **製品を良くする**ためにレビューをする
  - レビュイーに対して**攻撃しなくなる**

易しい
- 今後のレビューは**もっと気軽になる**かも
  - レビューという工程は無くなるかもしれない

レビューのカンファレンスを開催予定

開催時期：12月中旬

開催場所：東京都内

個人的な願望

- 様々なレビュー事例を聞きたい
- **「どのように考えてレビューを行っているのか」**について話したい

追加修正コストが発生するでしょう。

上に
はりつけ

伝えたい2つのこと

1. 隣の人はあなたが気付かなかったことを
共有していませんでしたか？

2. この例では何もプログラムを書いていません。
- 実装前に不具合を防ぐことができる例です。
- もしもこの時点で指摘できれば、
総コストが削減できるでしょう。

ご清聴ありがとうございました

（この発表も、背景にあるように沢山のレビューを経てお送りしました）