OSSユーザのための勉強会 #11

Sep 25th 2015

OSS を使用した開発のプロジェクト管理

SCSK 株式会社

R&Dセンター 技術開発部 技術戦略課

岩本 健 (IWAMOTO Takeshi)

SCSK

夢ある未来を、共に創る。

少し自己紹介をさせてください

Copyright(c) SCSK Corporation, ALL Rights Reserved.

岩本 健 (Iwamoto Takeshi) の仕事

SCSK

R&Dセンター 技術開発部 技術戦略課

これまでの仕事

- 人材育成 @人事
- 開発標準 @標準化
- 研究開発 @客先(某SI'er)
- 研究開発 @R&Dセンター
  - クラウドに関する研究
  - モバイルに関する研究
- 技術戦略 @R&Dセンター

興味・関心

- アジャイル開発
- アーキテクチャ

Copyright(c) SCSK Corporation, ALL Rights Reserved.

本日のアジェンダ

SCSK

1. OSS 活用の背景

2. OSS 活用で得たコト

3. まとめ

Copyright(c) SCSK Corporation, ALL Rights Reserved.

1. OSS 活用の背景

2. OSS 活用で得たコト

3. まとめ

Copyright(c) SCSK Corporation, ALL Rights Reserved.

研究開発における OSS 利用の背景

SCSK

- 研究開発業務の一つプロトタイプ開発
  - ✓ 新しい技術・デバイスなどの活用の仕方
  - ✓ 例えば、「IoT って仕事でどうやって使うの？」
- 研究開発組織なので、新しい仕様や技術をいち早く試すことができる。
  - ✓ 新技術の最も早い実装は OSS
  - ✓ JavaEE、Hadoop や Spark、Docker ・・・
- 無料で使える。検証段階なので助かる。

Copyright(c) SCSK Corporation, ALL Rights Reserved.

副次的な検証。せっかくなのであえて使う。

SCSK

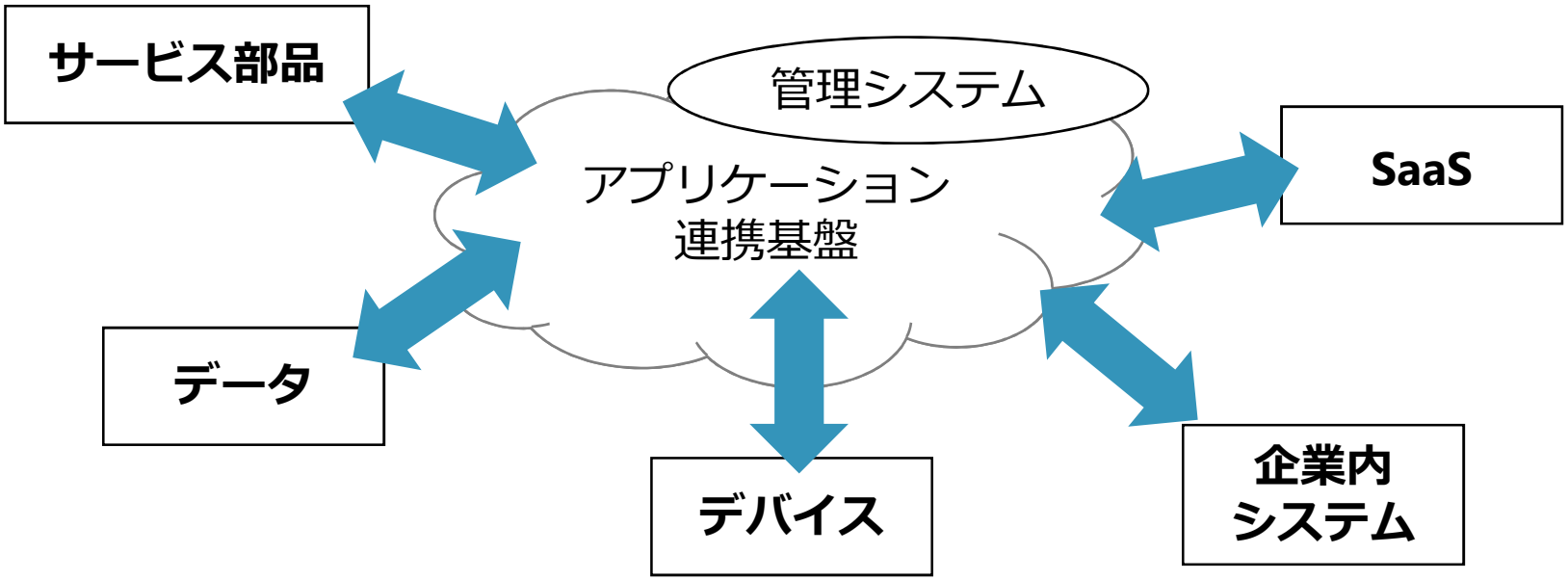
- いろんな開発プロセス・方法論ですすめる
  - ✓ アジャイル開発
  - ✓ 継続的インテグレーション (CI)
  - ✓ 継続的デリバリー
- 使ってみたい実装技術を影響の少ない部分で試す
  - ✓ JavaEE 7 (2013年当時)
  - ✓ NoSQL (Cassandra, MongoDB)
- 既存のOSSで代用できる部分は積極的に利用
  - ✓ ワークフロー

Copyright(c) SCSK Corporation, ALL Rights Reserved.

研究プロジェクトをご紹介

SCSK

- クラウド上でのアプリケーション連携基盤
  - ✓ 様々な外部のデータやサービスを連携するハブ
  - ✓ マルチテナント型 PaaS



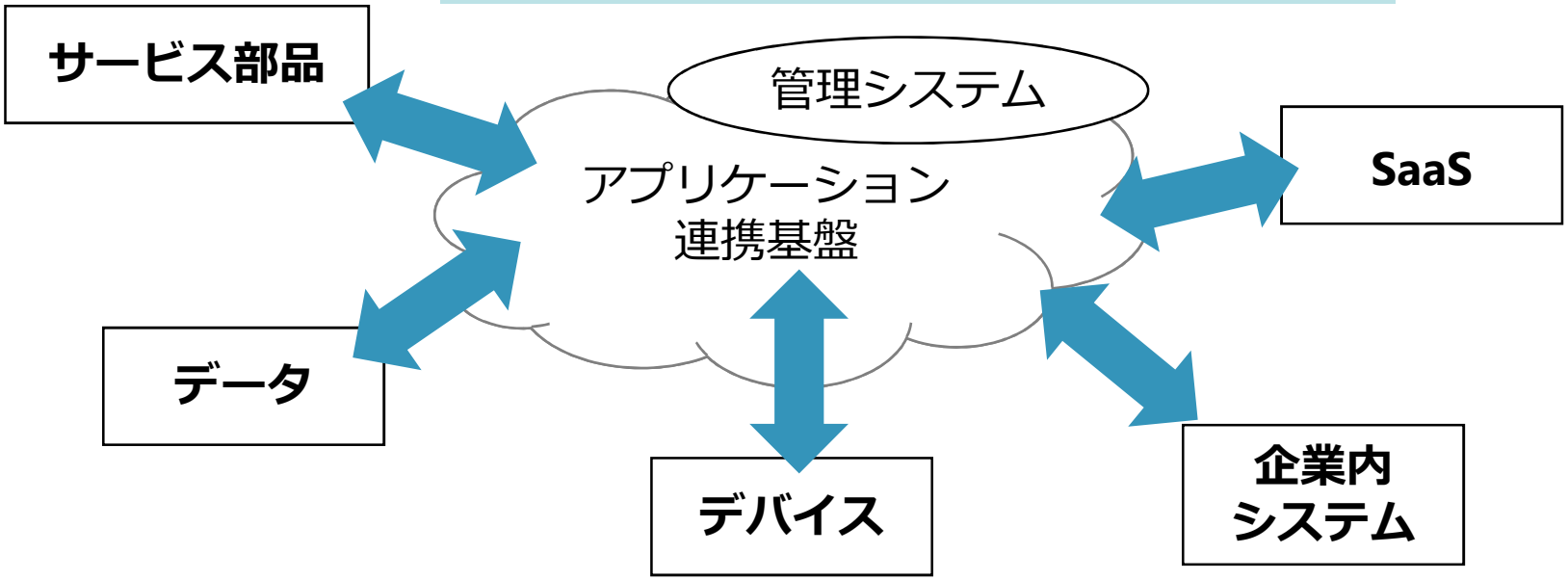
Copyright(c) SCSK Corporation, ALL Rights Reserved.

研究プロジェクトをご紹介

SCSK

- クラウド上でのアプリケーション連携基盤
  - ✓ 様々な外部のデータやサービスを連携するハブ
  - ✓ マルチテナント型 PaaS

・管理機能や周辺サービスを JavaEE で開発
・全体をアジャイル開発プロセスで遂行
・CI 環境を実験的に導入

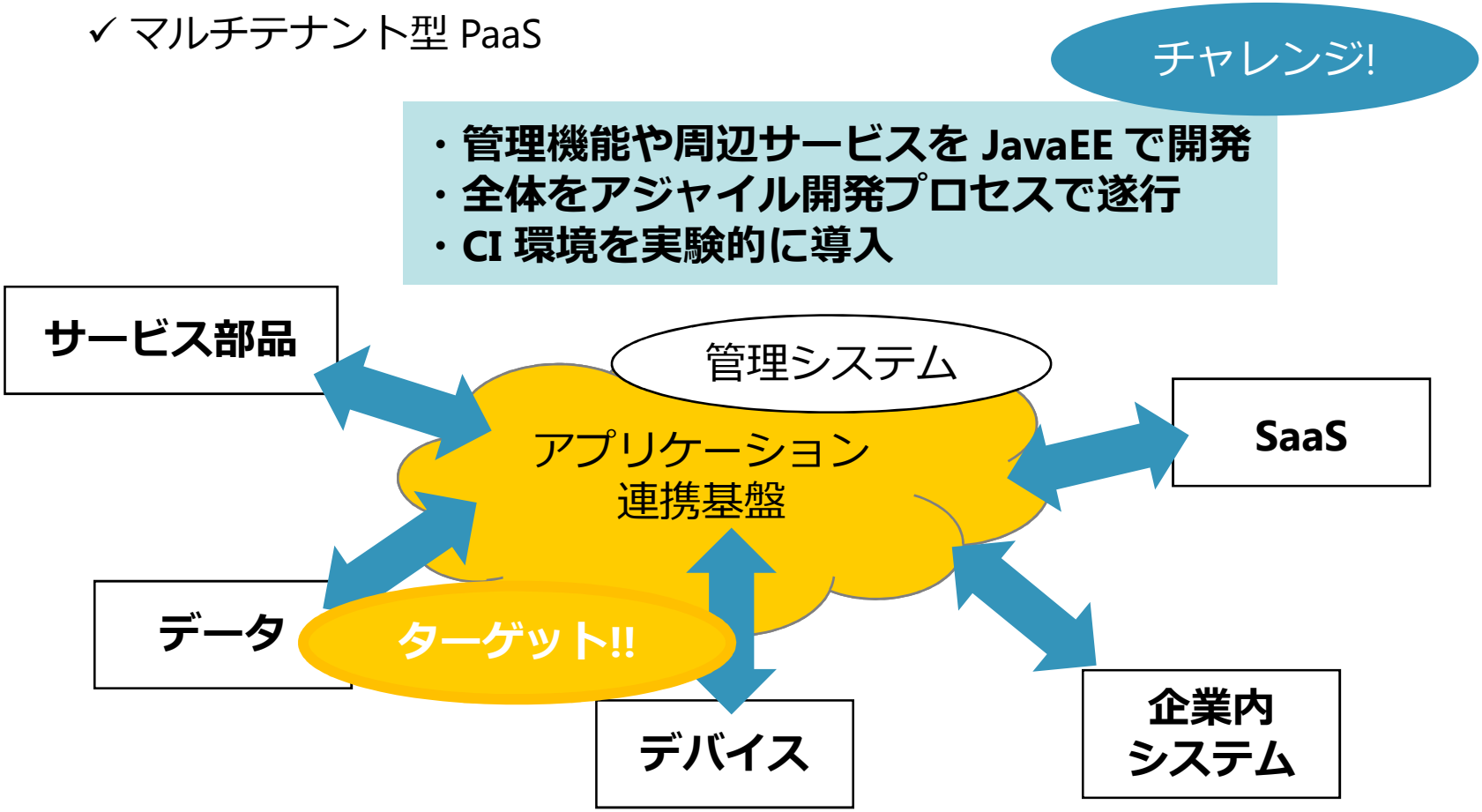


Copyright(c) SCSK Corporation, ALL Rights Reserved.

研究プロジェクトをご紹介

SCSK

- クラウド上でのアプリケーション連携基盤
  - ✓ 様々な外部のデータやサービスを連携するハブ
  - ✓ マルチテナント型 PaaS



Copyright(c) SCSK Corporation, ALL Rights Reserved.

プロジェクト管理の目標

SCSK

経験の少ないOSSや**技術・理論**を
試用してプロトタイプを実装し、

メインのターゲットである、
アプリケーション連携基盤の
アーキテクチャ及び
実現可能性の検証

を達成する。

Copyright(c) SCSK Corporation, ALL Rights Reserved.

今日のお話の主題

SCSK

経験の少ないOSSや**技術・理論**を
試用してプロトタイプを実装し、

メインのターゲットである、
アプリケーション連携基盤の
アーキテクチャ及び
実現可能性の検証

を達成する。

どう OSS と
向き合うか？？

Copyright(c) SCSK Corporation, ALL Rights Reserved.

1. OSS 活用の背景

2. OSS 活用で得たコト

3. まとめ

Copyright(c) SCSK Corporation, ALL Rights Reserved.

それぞれのシーン別に実体験を交えてご説明

SCSK

- いろんな開発プロセス・方法論ですすめる
  - ✓ アジャイル開発
  - ✓ 継続的インテグレーション (CI)
  - ✓ 継続的デリバリー
- 使ってみたい実装技術を影響の少ない部分で試す
  - ✓ JavaEE 7 (2013年当時)
  - ✓ NoSQL (Cassandra, MongoDB)
- 既存のOSSで代用できる部分は積極的に利用
  - ✓ ワークフロー

Copyright(c) SCSK Corporation, ALL Rights Reserved.

それぞれのシーン別に実体験を交えてご説明

SCSK

- いろんな開発プロセス・方法論ですすめる
  - ✓ **アジャイル開発**
  - ✓ **継続的インテグレーション (CI)**
  - ✓ **継続的デリバリー**
- 使ってみたい実装技術を影響の少ない部分で試す
  - ✓ **JavaEE 7 (2013年当時)**
  - ✓ NoSQL (Cassandra, MongoDB)
- 既存のOSSで代用できる部分は積極的に利用
  - ✓ ワークフロー

Copyright(c) SCSK Corporation, ALL Rights Reserved.

それぞれのシーン別に実体験を交えてご説明

SCSK

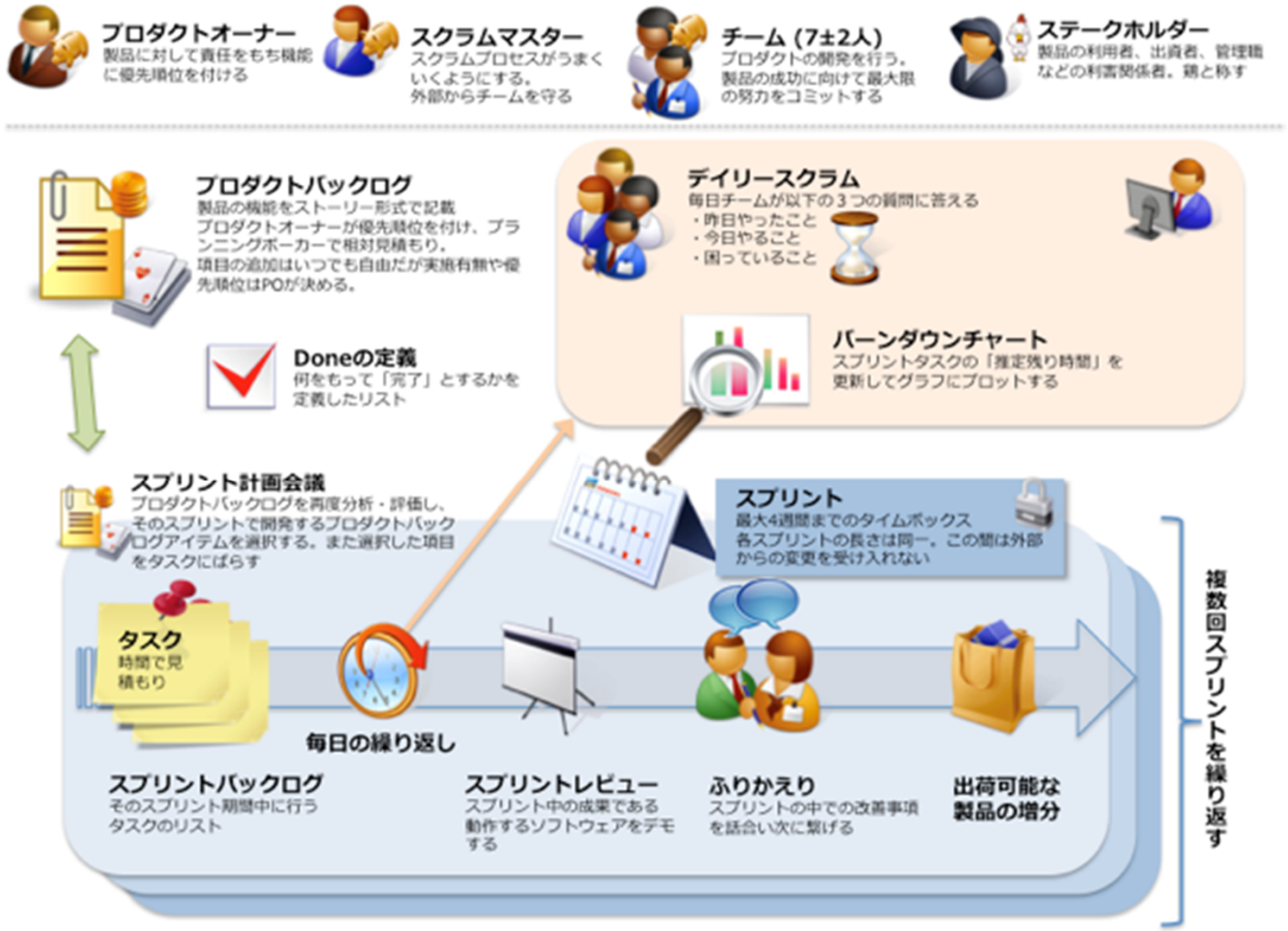
- いろんな開発プロセス・方法論ですすめる
  - ✓ **アジャイル開発**
  - ✓ **継続的インテグレーション (CI)**
  - ✓ **継続的デリバリー**
- 使ってみたい実装技術を影響の少ない部分で試す
  - ✓ **JavaEE 7 (2013年当時)**
  - ✓ NoSQL (Cassandra, MongoDB)
- 既存のOSSで代用できる部分は積極的に利用
  - ✓ **ワークフロー**

Copyright(c) SCSK Corporation, ALL Rights Reserved.

プロセスには「Scrum」を活用

SCSK

アジャイル開発の開発プロセスを定義した中で最もポピュラーな Scrumを活用



©2011 Ryuzee. All rights reserved. http://www.ryuzee.com/

Copyright(c) SCSK Corporation, ALL Rights Reserved.

Scrum ご存知の方？

Copyright(c) SCSK Corporation, ALL Rights Reserved.

では、

SCSK

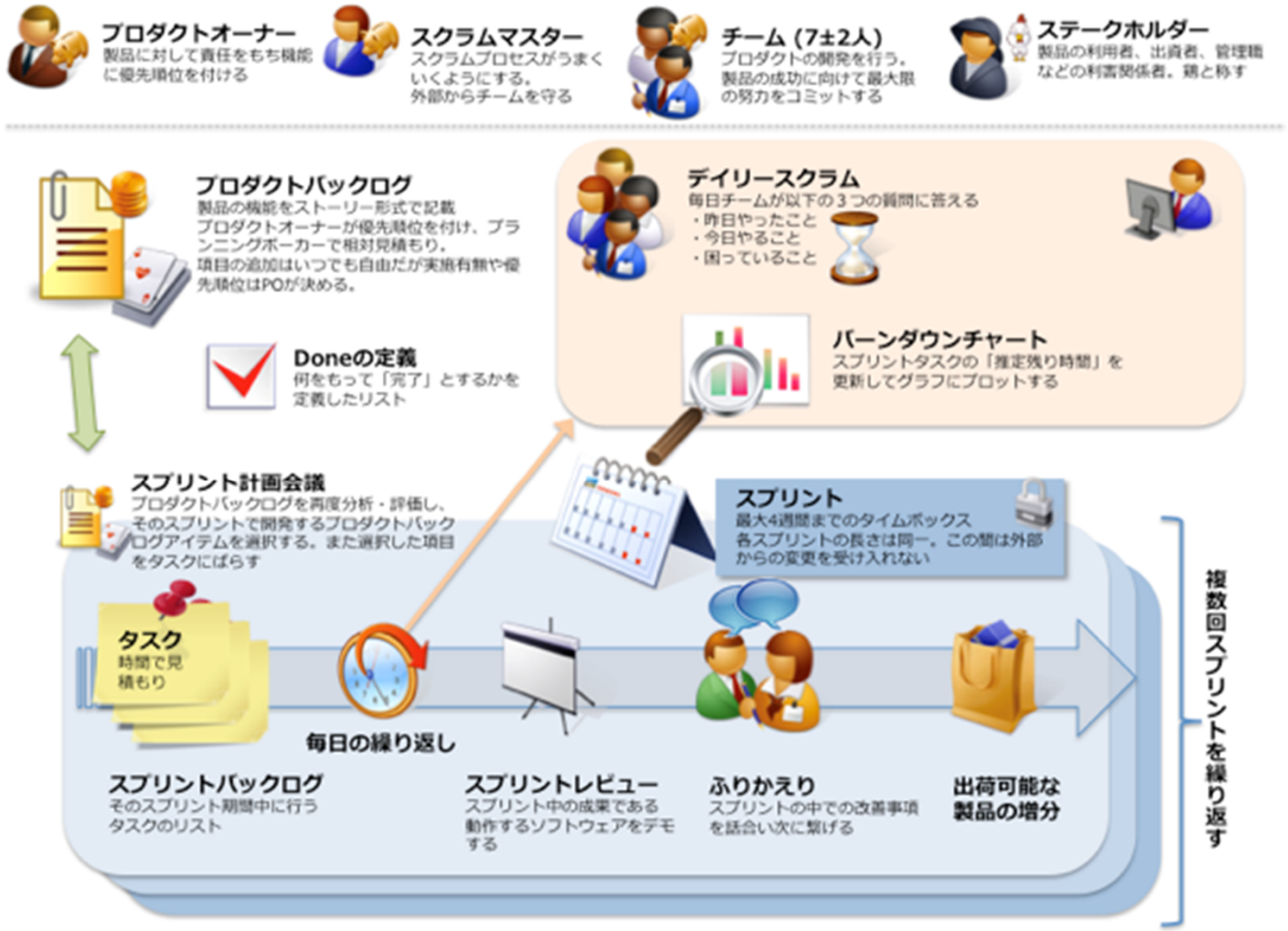
少しだけ説明します

Copyright(c) SCSK Corporation, ALL Rights Reserved.

アジャイル開発プロセス 「Scrum」

SCSK

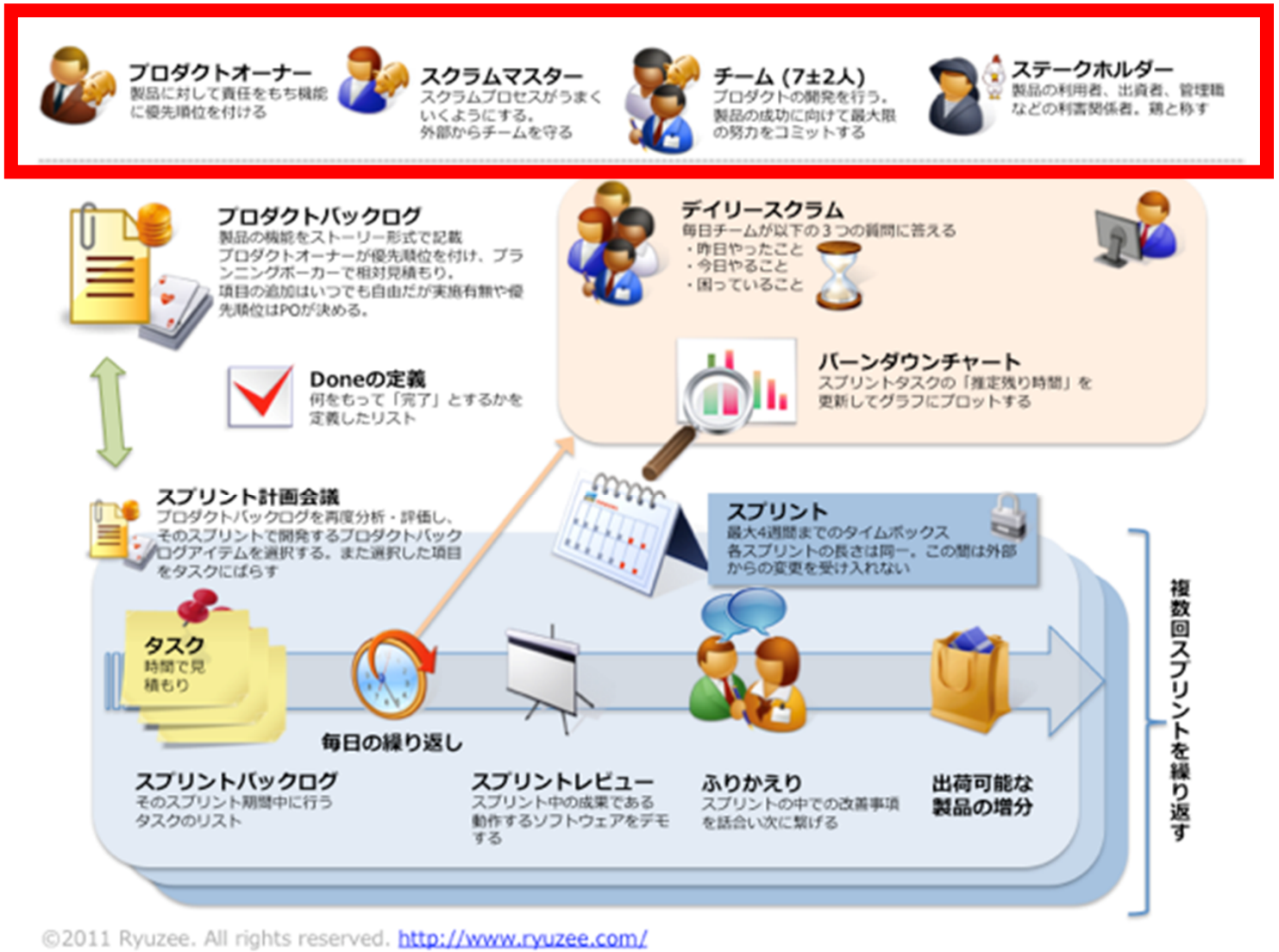
アジャイル開発の開発プロセスを定義した中で最もポピュラーな Scrumを活用



Copyright(c) SCSK Corporation, ALL Rights Reserved.

Scrum の役割 ４つ

アジャイル開発の開発プロセスを定義した中で最もポピュラーな Scrumを活用

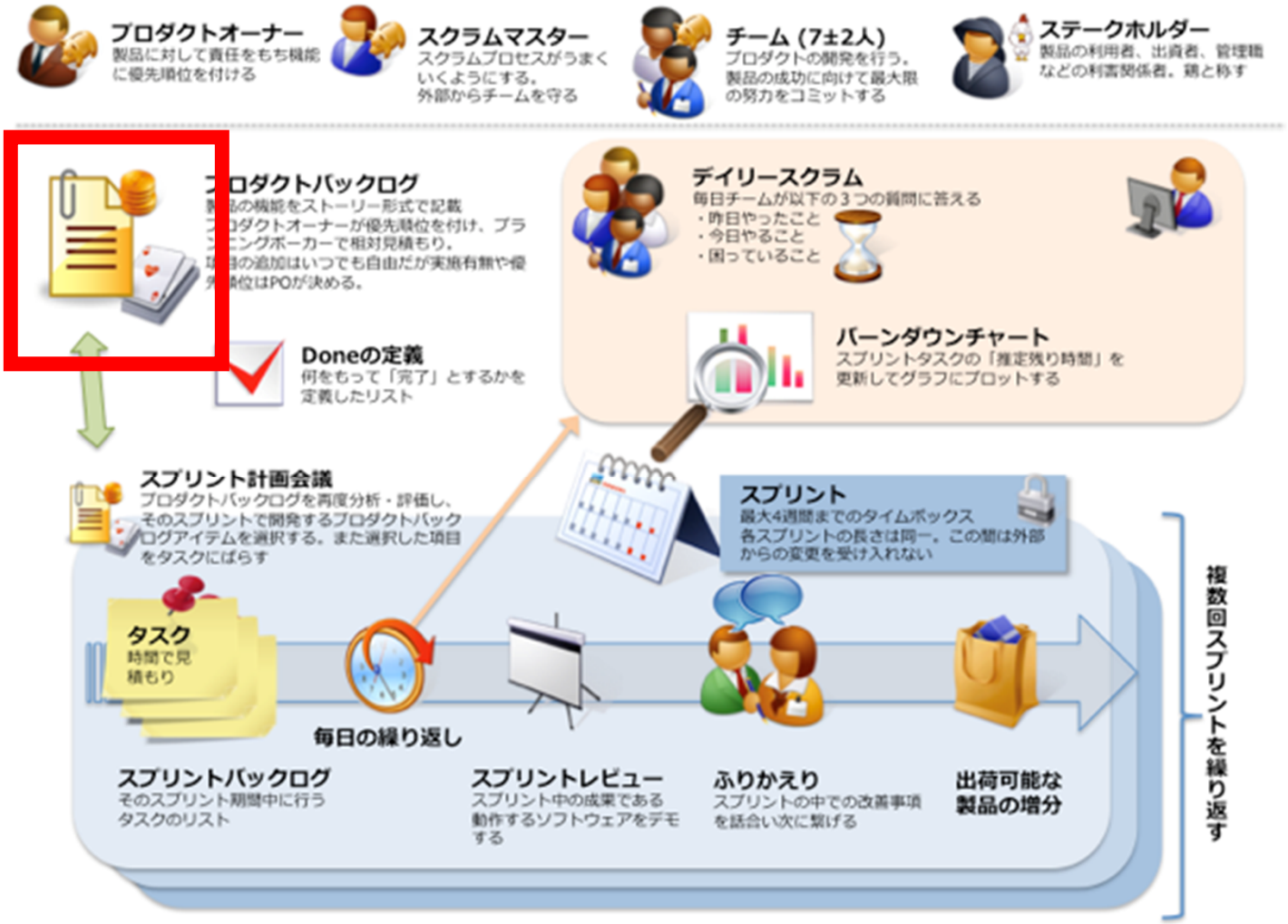


©2011 Ryuzee. All rights reserved. http://www.ryuzee.com/

Copyright(c) SCSK Corporation, ALL Rights Reserved.

Scrum の要件リスト

アジャイル開発の開発プロセスを定義した中で最もポピュラーな Scrumを活用



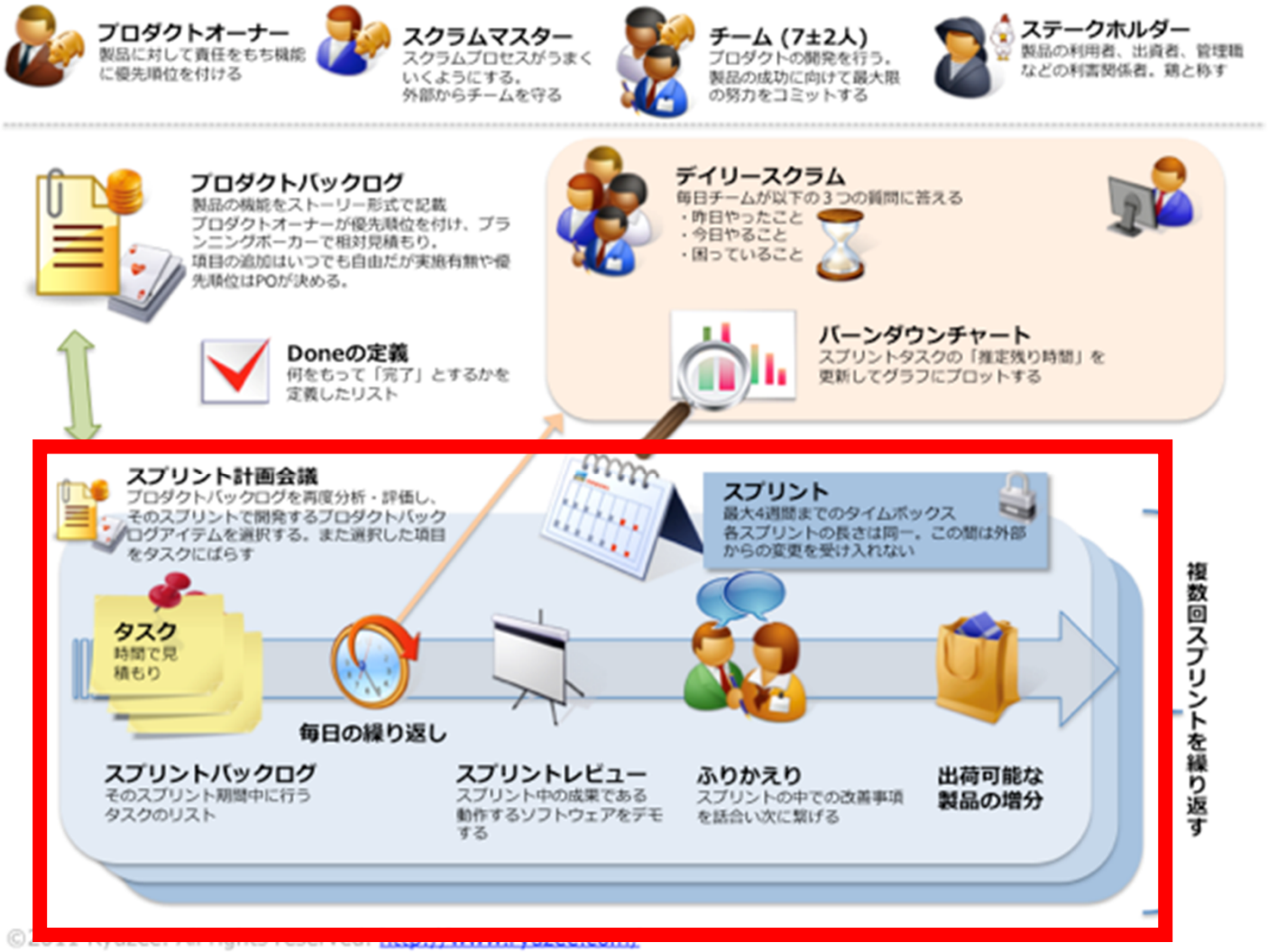
©2011 Ryuzee. All rights reserved. http://www.ryuzee.com/

Copyright(c) SCSK Corporation, ALL Rights Reserved.

Scrum の開発単位（2 週間~4 週間）

SCSK

アジャイル開発の開発プロセスを定義した中で最もポピュラーな Scrumを活用



Copyright(c) SCSK Corporation, ALL Rights Reserved.

Scrum の特徴は一定期間で改善を繰り返す

SCSK

- 2W～4W単位で改善を繰り返す
  - ✓ 改善例
    - 開発者それぞれで実装イメージが異なっていた。「ああしようと思ってた」「こうしようと思ってた」が後から噴出。
    - 新しいバックログを開始するときはまず声をかける
    - まずはスケルトンコードとテストコードを書く。
      ⇒ 実装方針を決める。(最初は口頭で、その次はスケルトン、最後はスケルトンコードをペアプロするように改善)
- 見積もりの制度を上げる
  - ✓繰り返し開発していると、チームの力が計測できる
  - ✓毎回見積もりと実績をとり、見積もり精度も改善。

Copyright(c) SCSK Corporation, ALL Rights Reserved.

改善を繰り返すためには・・・

- 2W～4W単位で改善を繰り返す
  - ✓ 改善例
    - 開発者それぞれで実装イメージが異なっていた。「ああしようと思ってた」「こうしようと思ってた」が後から噴出。
    - まずはスケルトンコードというスケルトンコードを書く。
      ⇒ 実装方針を決める。(最初は口頭で、その次はスケルトン、最後はスケルトンコードをペアプロするように改善)

計測の精度が重要

- 見積もりの制度を上げる
  - ✓繰り返し開発していると、チームの力が計測できる
  - ✓毎回見積もりと実績をとり、見積もり制度も改善。

Copyright(c) SCSK Corporation, ALL Rights Reserved.

Redmine の導入

SCSK

プロジェクト管理・カンバン

Apache maven Jenkins

vmware

ツールを導入してデータを収集し、
計測の制度を上げつつ省力化しよう！

ORACLE VM VirtualBox

VAGRANT

個人開発環境

GitLab

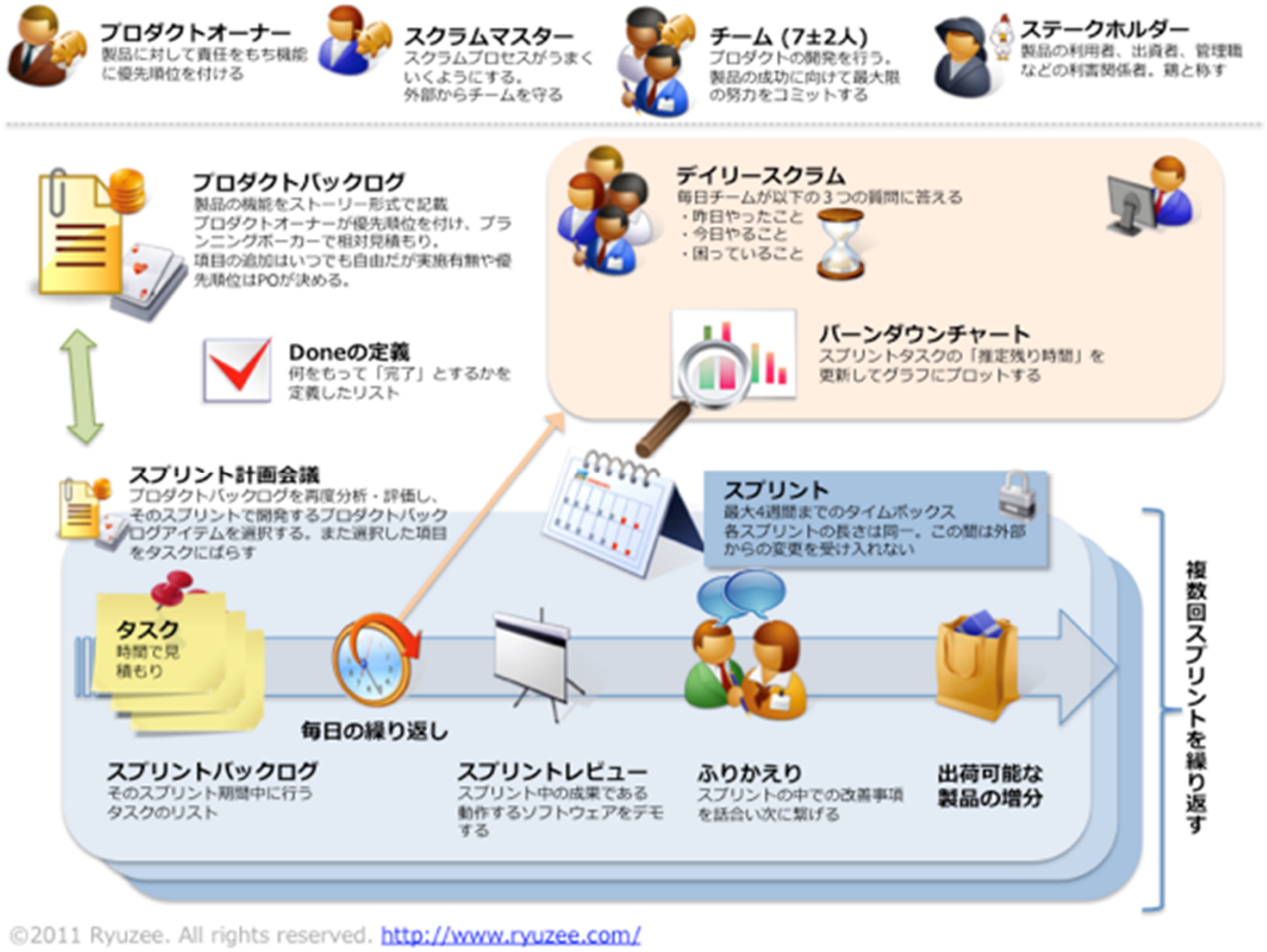
git

ソースコード管理

Copyright(c) SCSK Corporation, ALL Rights Reserved.

何を計測するか？

アジャイル開発の開発プロセスを定義した中で最もポピュラーな Scrumを活用



Copyright(c) SCSK Corporation, ALL Rights Reserved.

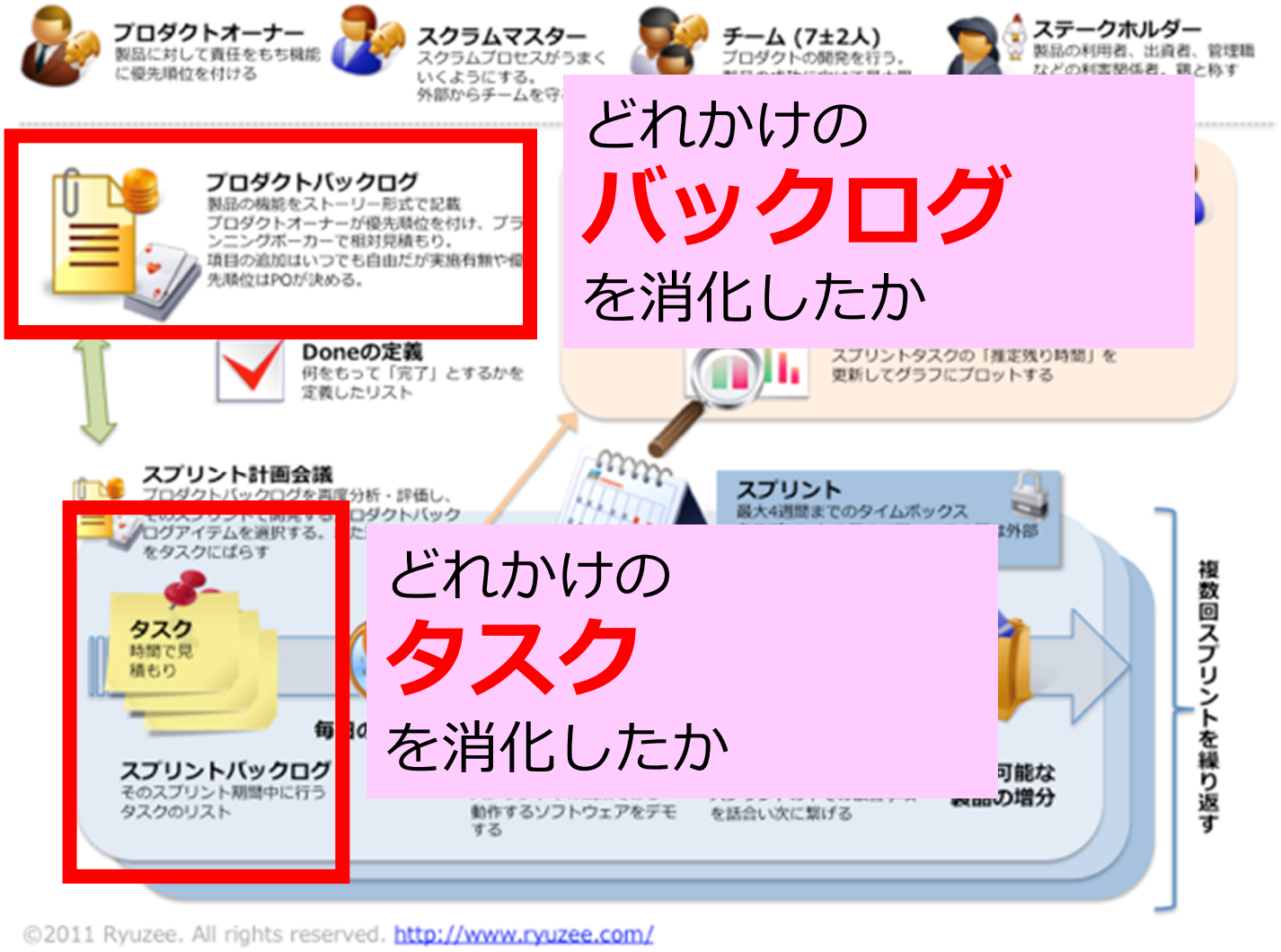
何を計測するか？

アジャイル開発の開発プロセスを定義した中で最もポピュラーな Scrumを活用

Copyright(c) SCSK Corporation, ALL Rights Reserved.

何を計測するか？

アジャイル開発の開発プロセスを定義した中で最もポピュラーな Scrumを活用



Copyright(c) SCSK Corporation, ALL Rights Reserved.

Redmineで不足する機能を補う

SCSK

- Redmineだけでは、Scrum で必要な情報を収集できない(できるが面倒)

- Scrum 開発を実践するようなプラグインはいくつか開発されている。

- Backlogs プラグインというツールが良さそうだ！

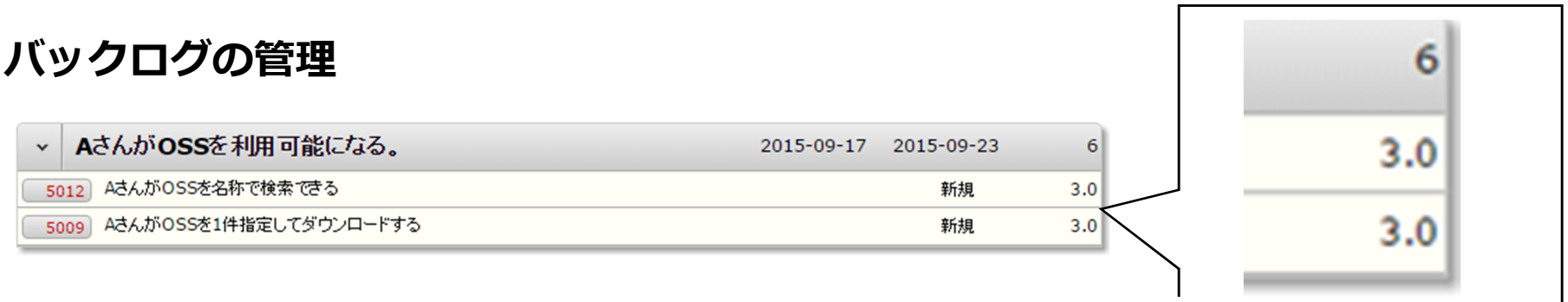
  ✓特に実施検証せず

  ✓ドキュメントやWebの情報で機能やできることを確認

Copyright(c) SCSK Corporation, ALL Rights Reserved.

プラグイン 「Backlogs」 を利用

バックログの管理



バックログにひもづく タスクの管理

Copyright(c) SCSK Corporation, ALL Rights Reserved.

タスクの総時間 = 予定工数

タスク

スプリントの **予定工数(時間)**

Copyright(c) SCSK Corporation, ALL Rights Reserved.

イメージして頂けましたか？

Copyright(c) SCSK Corporation, ALL Rights Reserved.

それぞれのシーン別に実体験を交えてご説明

SCSK

- いろんな開発プロセス・方法論ですすめる
  - ✓ **アジャイル開発**
  - ✓ **継続的インテグレーション (CI)**
  - ✓ **継続的デリバリー**
- 使ってみたい実装技術を影響の少ない部分で試す
  - ✓ **JavaEE 7 (2013年当時)**
  - ✓ NoSQL (Cassandra, MongoDB)
- 既存のOSSで代用できる部分は積極的に利用
  - ✓ **ワークフロー**

Copyright(c) SCSK Corporation, ALL Rights Reserved.

数字が合わない

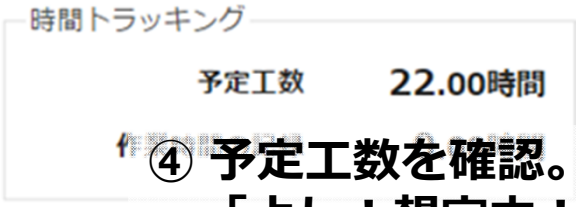
Copyright(c) SCSK Corporation, ALL Rights Reserved.

詳細タスク計画時の流れ

① タスクを切って、作業時間を設定。

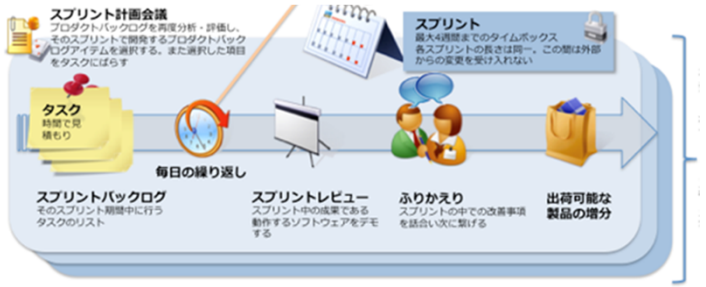
② 「画面実装」タスクの見積もり時間を変更

③ 作業時間の見積もり完了



④ 予定工数を確認。「よし！想定内！！」

⑤ スプリント(開発) 開始！！



Copyright(c) SCSK Corporation, ALL Rights Reserved.

？？？

① タスクを切って、作業時間を設定。

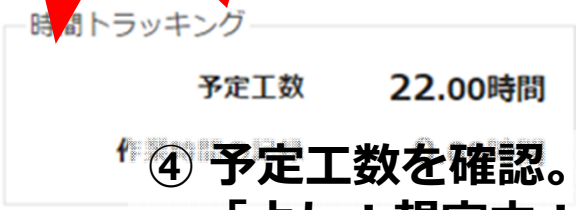
なんか
作業時間が多い

② 「画面実装」タスクの
見積もり時間を変更

③ 作業時間の
見積もり完了

⑤ スプリント(開発) 開始！！

最初の予定工数
おかしな気がする



④ 予定工数を確認。
「よし！想定内！！」

Copyright(c) SCSK Corporation, ALL Rights Reserved.

よく見てみるとタスク時間の総和が違う

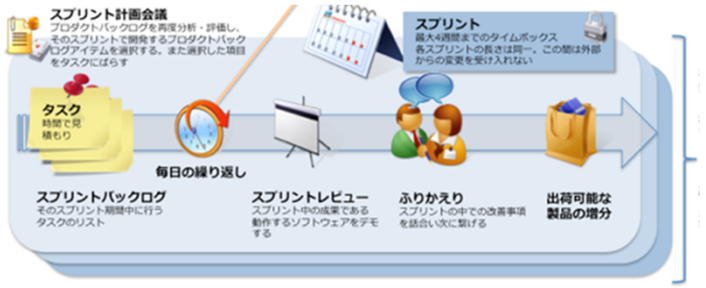
SCSK

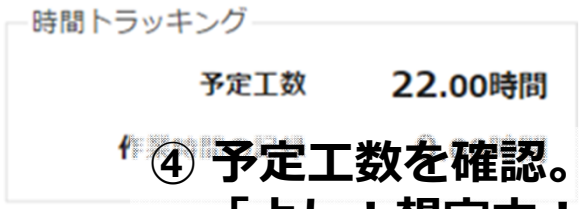
① タスクを切って、作業時間を設定。

② 「画面実装」タスクの見積もり時間を変更

③ 作業時間の見積もり完了

⑤ スプリント(開発) 開始！！





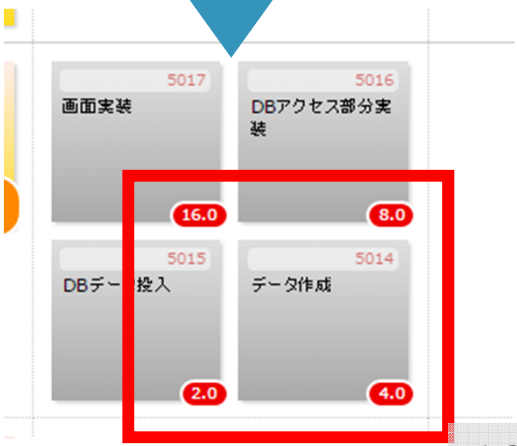
④ 予定工数を確認。「よし！想定内！！」

Copyright(c) SCSK Corporation, ALL Rights Reserved.

よく見てみるとタスク時間の総和が違う

① タスクを切って、作業時間を設定。

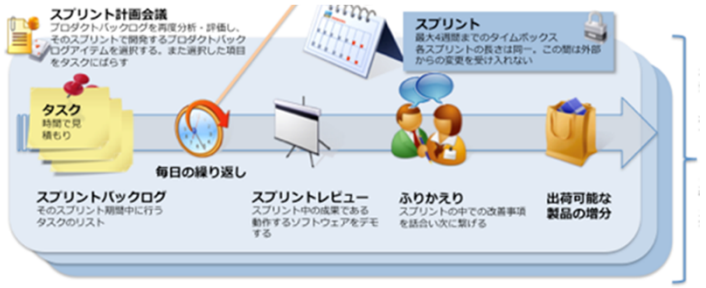
② 「画面実装」タスクの見積もり時間を変更

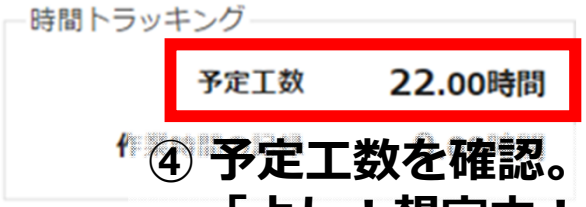


③ 作業時間の見積もり完了

16.0 + 8.0 + 2.0 + 4.0 = 30.0

⑤ スプリント(開発) 開始！！





④ 予定工数を確認。
「よし！想定内！！」

Copyright(c) SCSK Corporation, ALL Rights Reserved.

それぞれの数字の意味が合っていない

SCSK

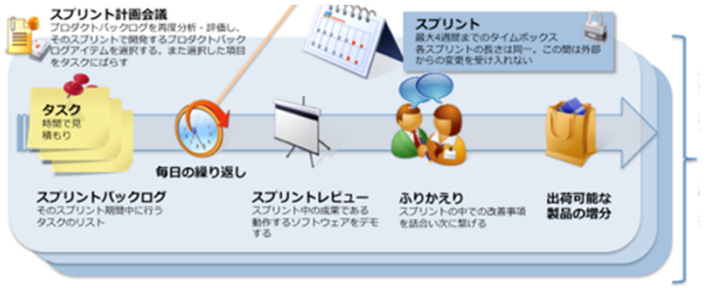
① タスクを切って、作業時間を設定。

Backlogs
プラグインの
世界

② 「画面実装」タスクの
見積もり時間を変更

③ 作業時間の
見積もり完了

⑤ スプリント(開発) 開始！！



Redmine
ベース機能の
世界

④ 予定工数 22.00時間
「よし！想定内！！」

Copyright(c) SCSK Corporation, ALL Rights Reserved.

こんなややこしいなら・・・

SCSK

- Excel のほうがいいのでは？自分で計算式を書いたほうが良いのでは。
- Redmineや Backlogs プラグインはブラックボックスで実際に何をやっているのかは見えにくい。

Copyright(c) SCSK Corporation, ALL Rights Reserved.

いいコトだってあるんです。

SCSK

- Webシステムなので複数人の入力インタフェースとしては優れている。
- データの保管、変更履歴の取得。
- プラグインによる機能強化
- Scrum に特有な機能を多数備えている。
  - ✓ タスクのカンバン表示など。
  - ✓ Scrum に必要な様々な値を自動で取得し計算

Copyright(c) SCSK Corporation, ALL Rights Reserved.

ツール系選定時の反省点

SCSK

「プラグインを入れればできる」は信用しすぎるな！！

Backlogsプラグインは
Redmine が提供するデータ項目をうまく使おうとしたあまり
逆に理解や使い方が難しくなった。(バグではない)

あまり重要でないことに見えるが、計測を重んじる
Scrumとしては正しい工数を取得したい。

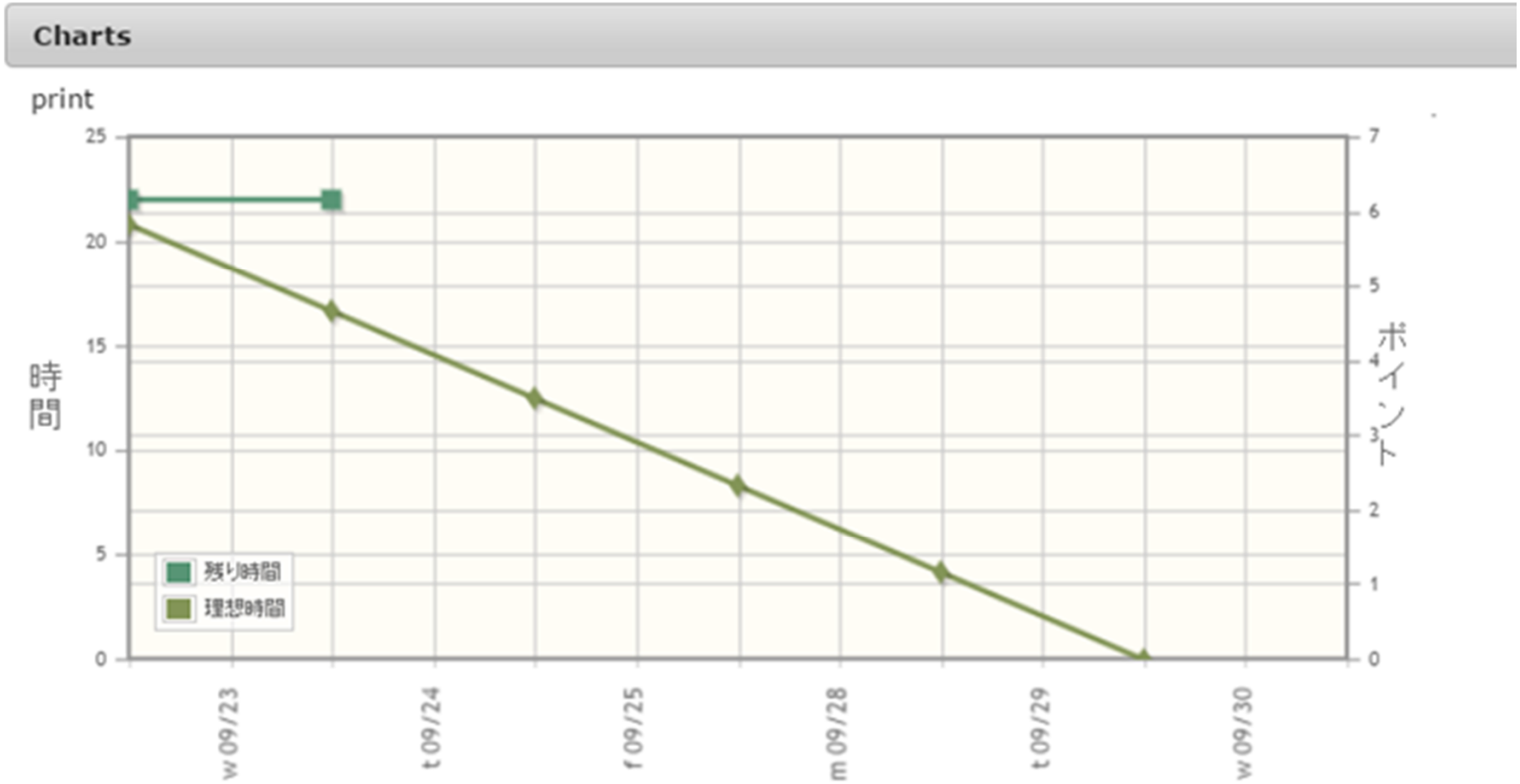
Webの情報、思い込みで使用を開始するのではなく、
いったんプロジェクトのウォークスルーを実施し、
各値の有効性などを検証すべきであった。

Copyright(c) SCSK Corporation, ALL Rights Reserved.

プラグインではなくアナログツールを併用した方が良い時もある

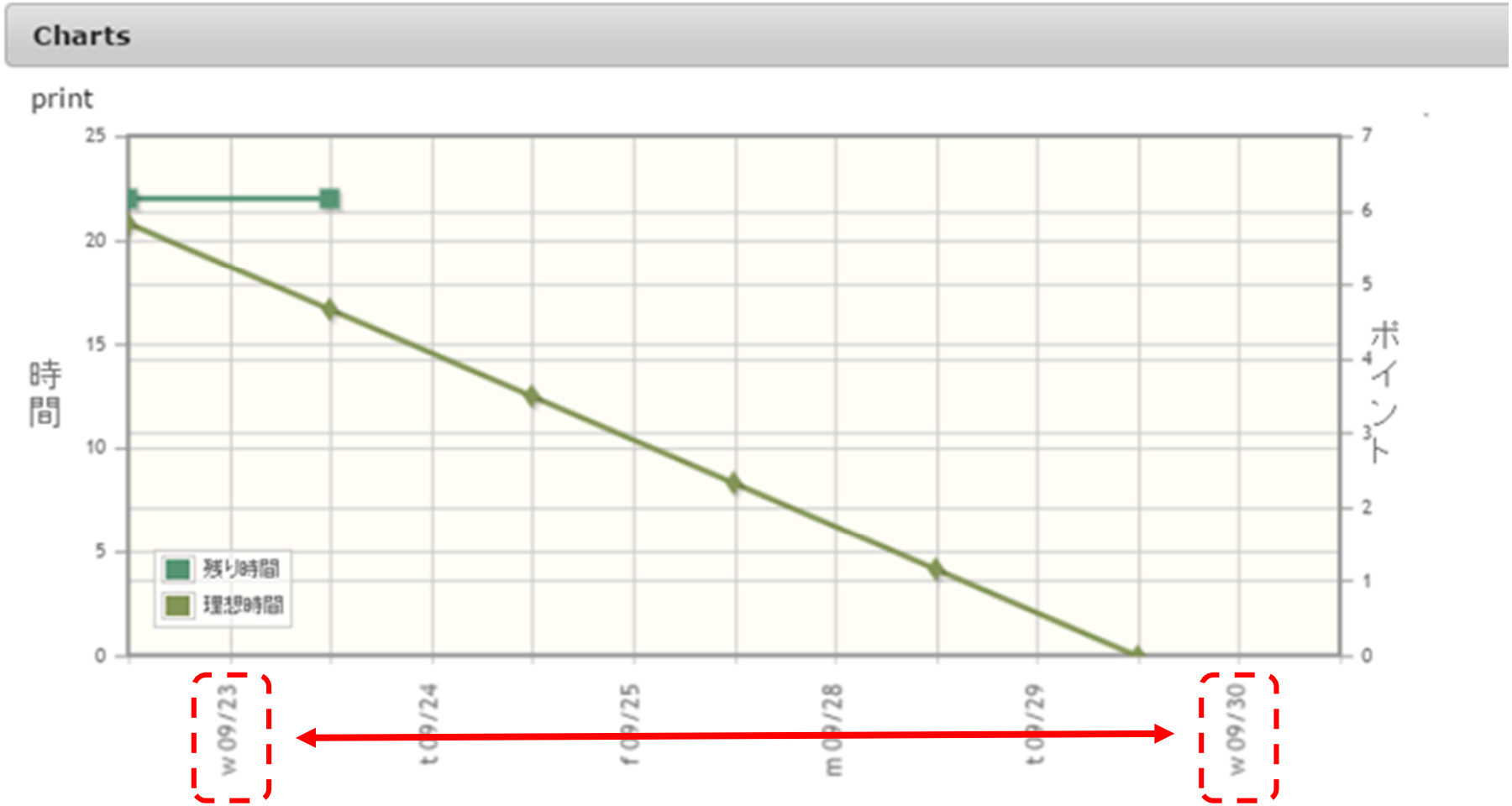
Copyright(c) SCSK Corporation, ALL Rights Reserved.

Scrum での日々の進捗管理



Copyright(c) SCSK Corporation, ALL Rights Reserved.

Scrum での日々の進捗管理



スプリントの期間

Copyright(c) SCSK Corporation, ALL Rights Reserved.

Scrum での日々の進捗管理

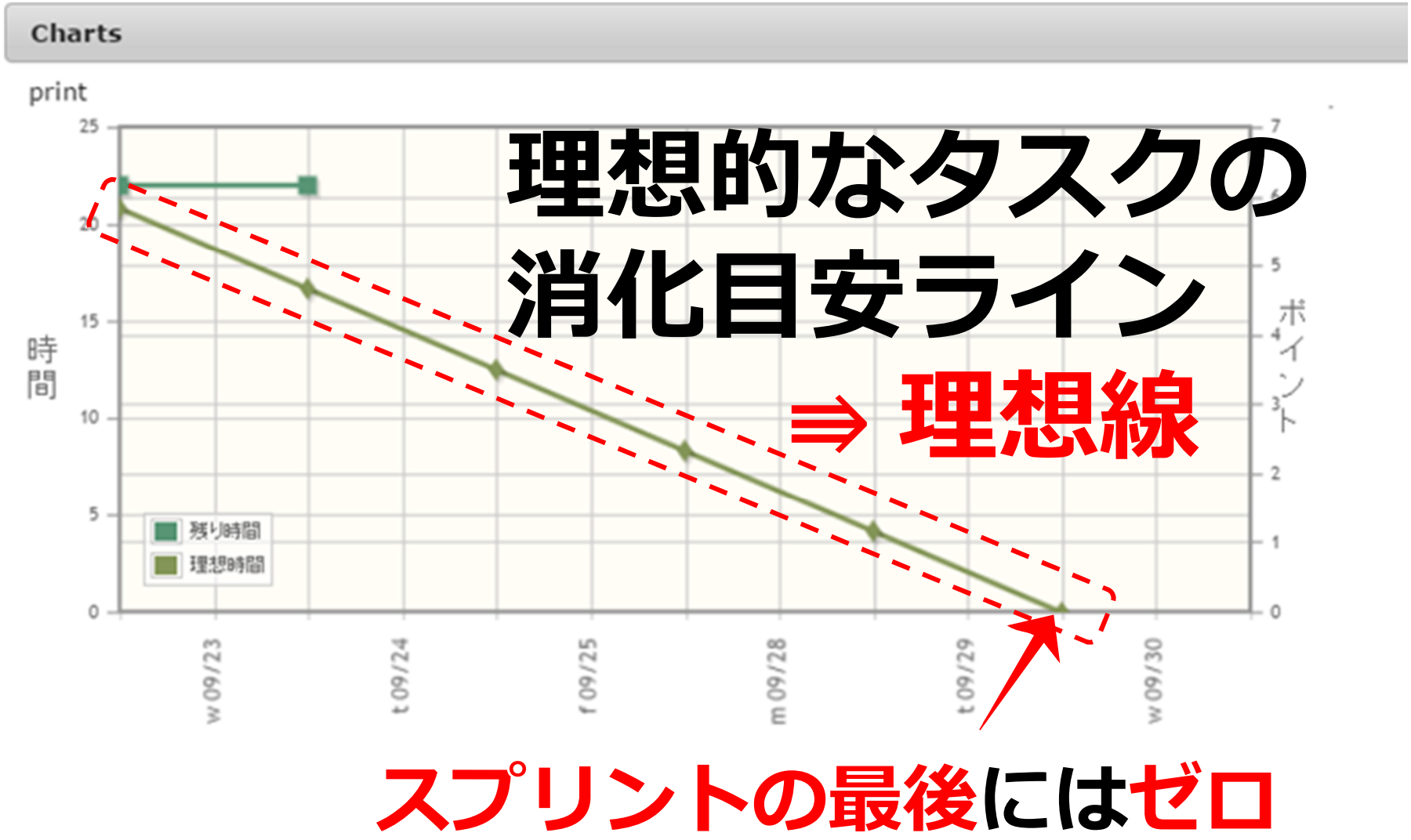
作業時間

Copyright(c) SCSK Corporation, ALL Rights Reserved.

Scrum での日々の進捗管理

Copyright(c) SCSK Corporation, ALL Rights Reserved.

バーンダウンチャートで日々の進捗管理



Copyright(c) SCSK Corporation, ALL Rights Reserved.

Scrum での日々の進捗管理

Copyright(c) SCSK Corporation, ALL Rights Reserved.

バーンダウンチャートで日々の進捗管理

Copyright(c) SCSK Corporation, ALL Rights Reserved.

バーンダウンチャートで日々の進捗管理

Copyright(c) SCSK Corporation, ALL Rights Reserved.

バーンダウンチャートで日々の進捗管理

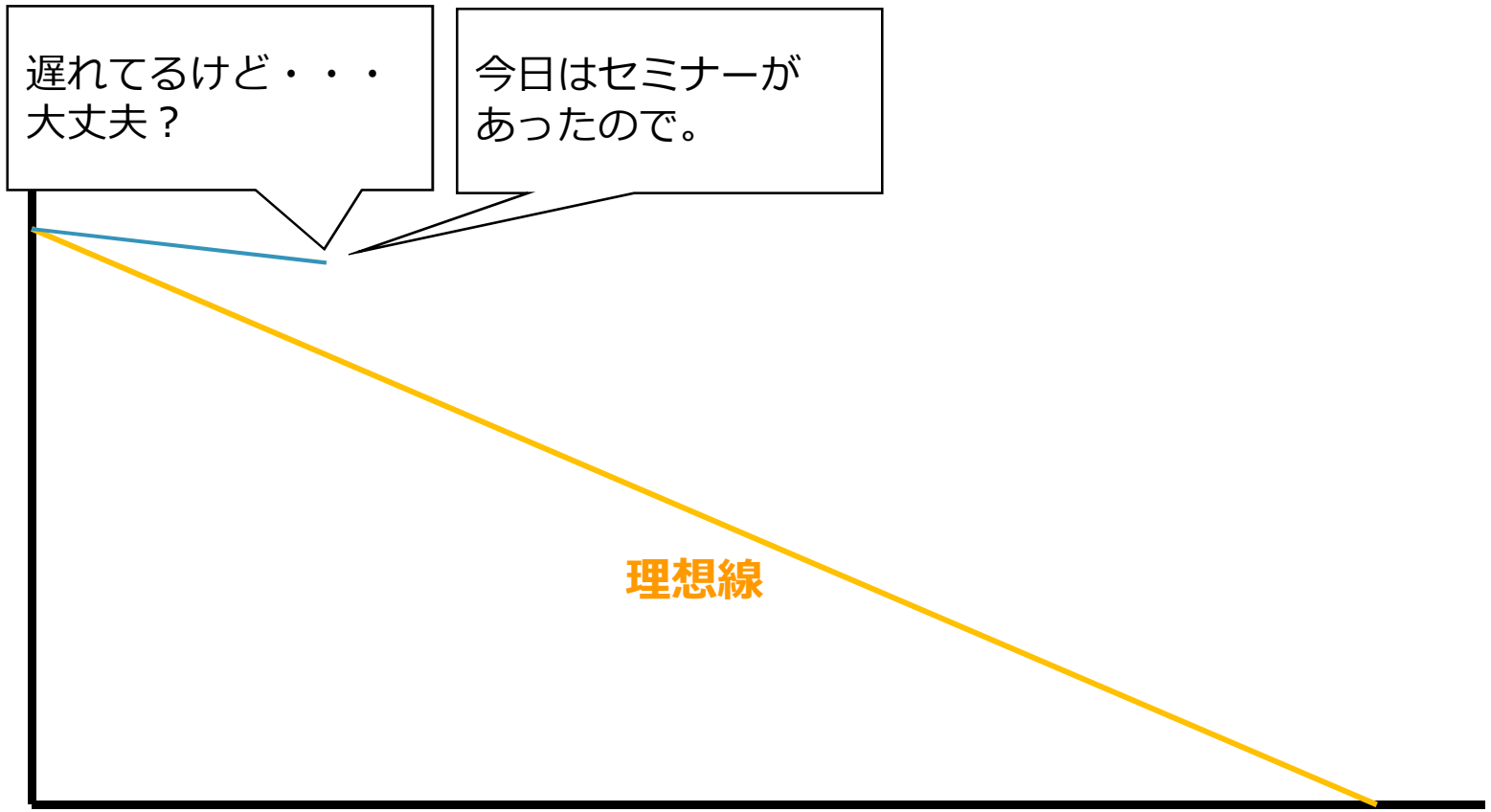
Backlogs プラグインにバーンダウンチャート機能を使ってみた。

Copyright(c) SCSK Corporation, ALL Rights Reserved.

実際のバーンダウンチャートの例

SCSK

技術的な問題に加え、会社の事務作業などが入る。

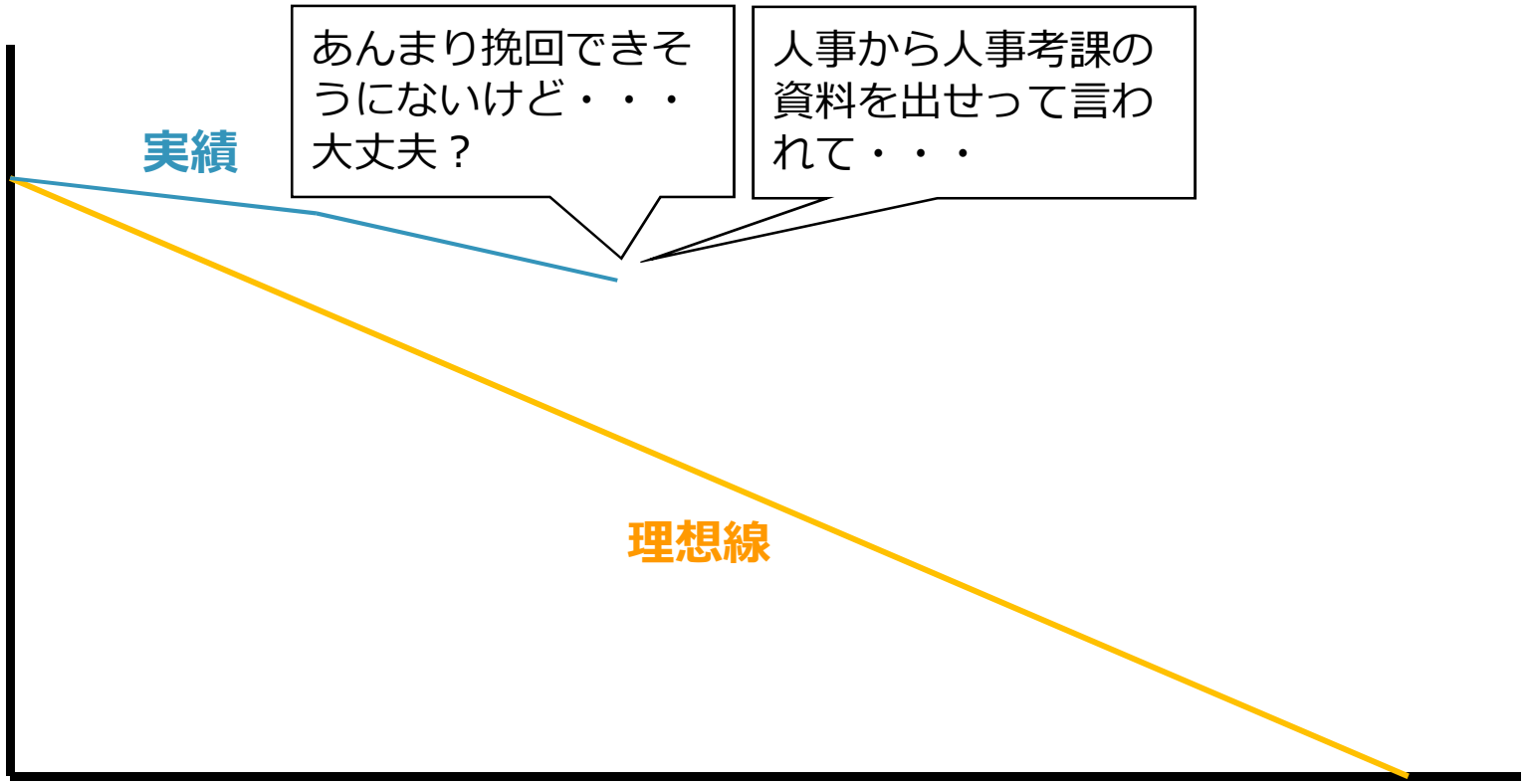


Copyright(c) SCSK Corporation, ALL Rights Reserved.

実際のバーンダウンチャートの例

SCSK

技術的な問題に加え、会社の事務作業などが入る。

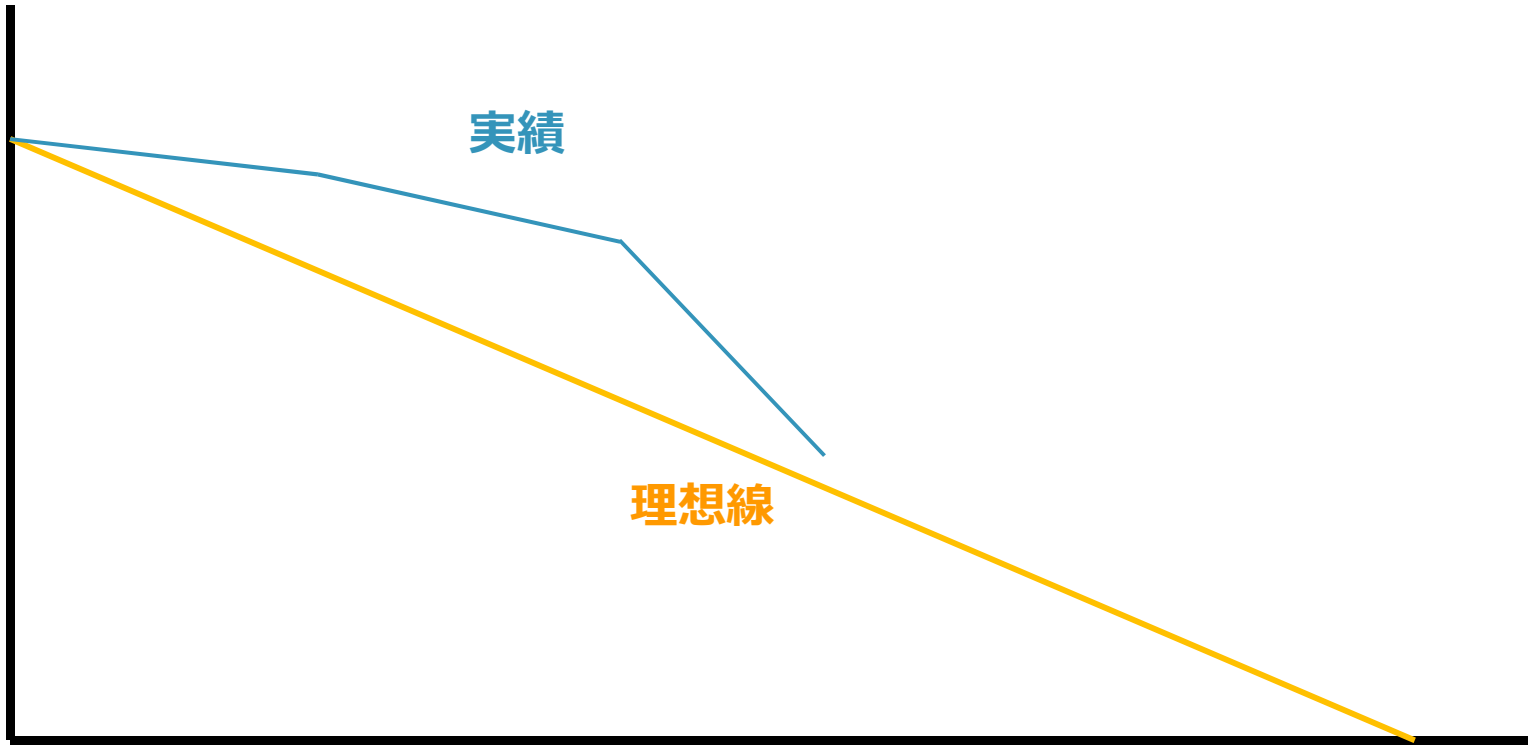


Copyright(c) SCSK Corporation, ALL Rights Reserved.

実際のバーンダウンチャートの例

SCSK

技術的な問題に加え、会社の事務作業などが入る。

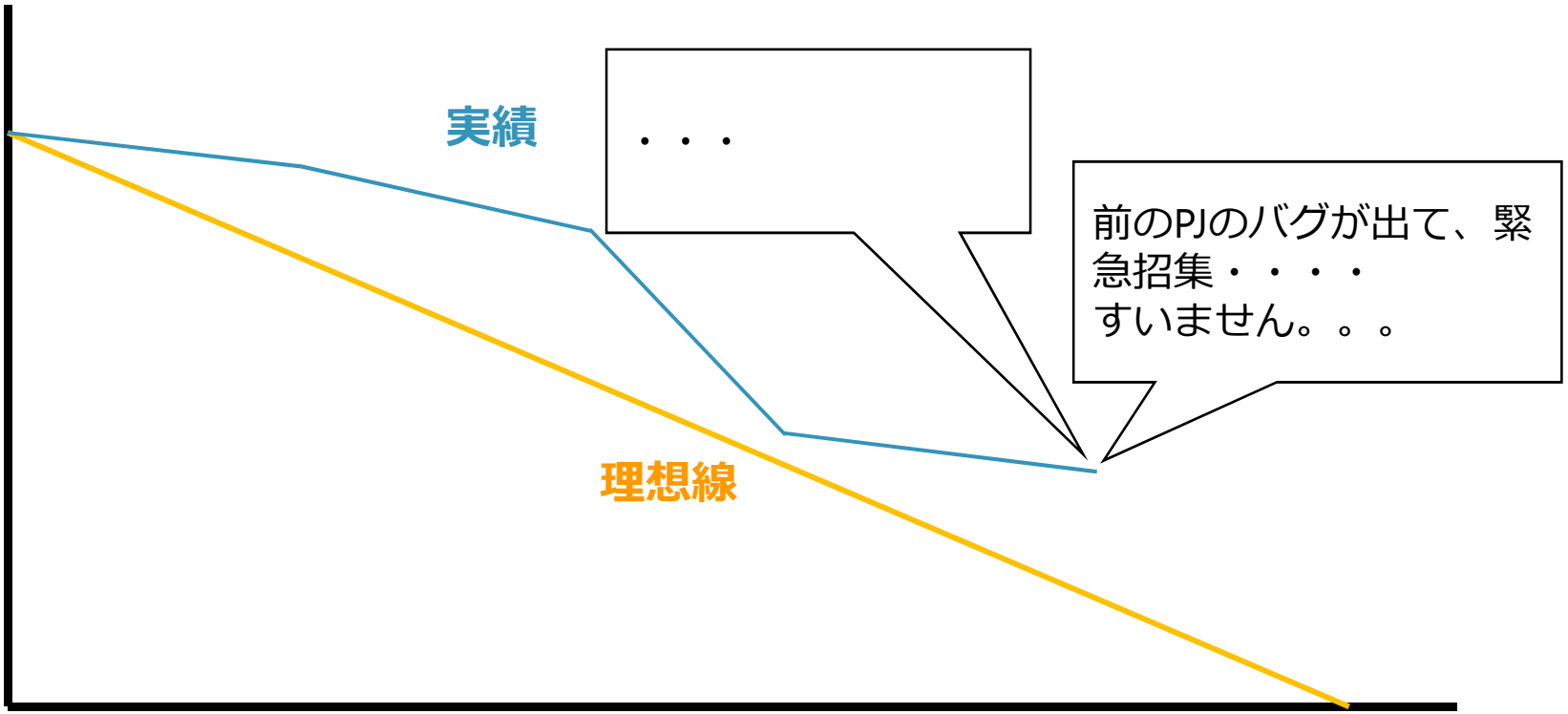


Copyright(c) SCSK Corporation, ALL Rights Reserved.

実際のバーンダウンチャートの例

SCSK

技術的な問題に加え、会社の事務作業などが入る。

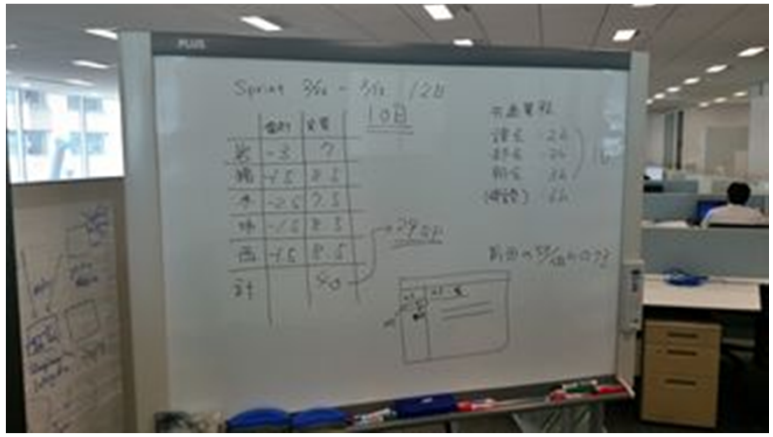


Copyright(c) SCSK Corporation, ALL Rights Reserved.

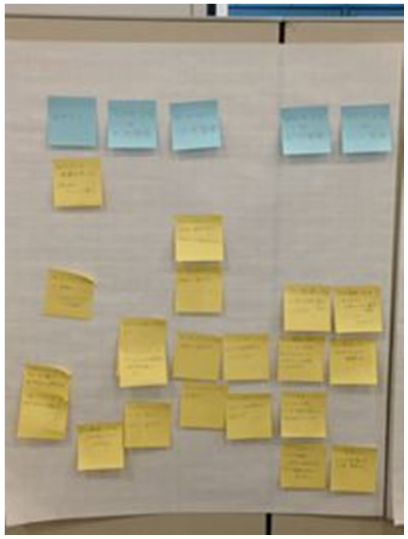
Redmineと アナログツールの併用

SCSK

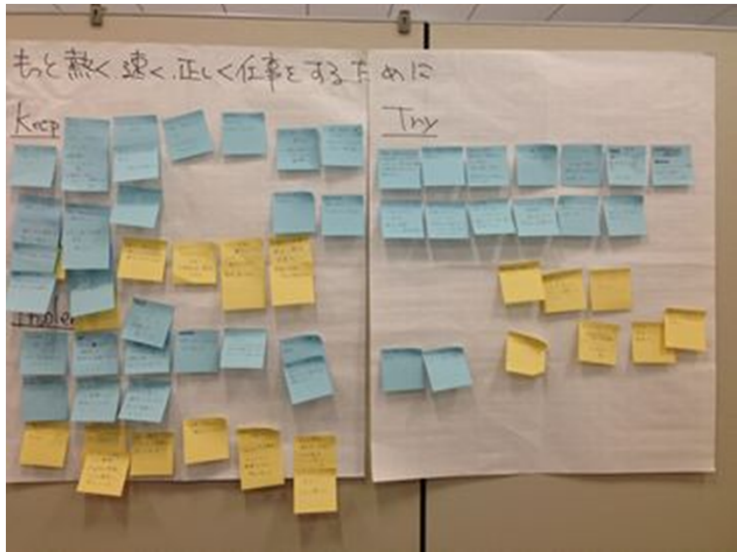
ベロシティ計測



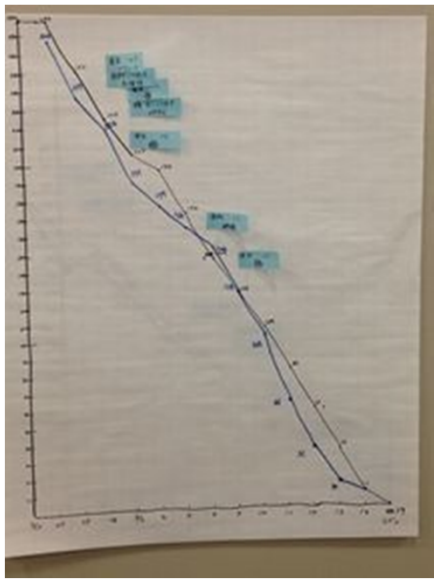
ユーザーストーリマッピング



レトロスペクティブ (プラクティス：KPT)



バーンダウンチャート

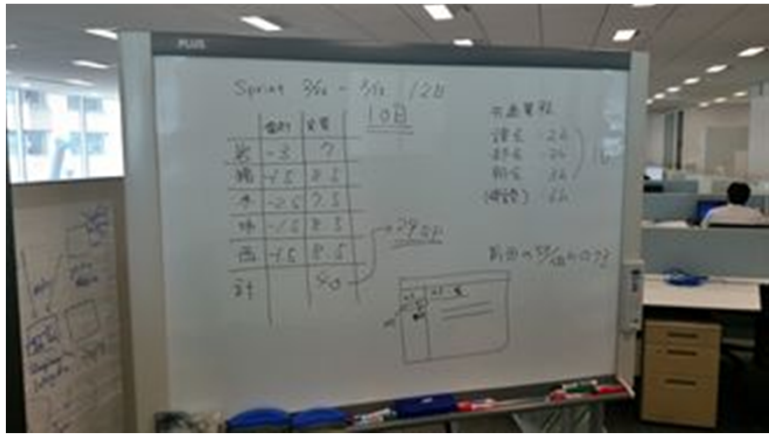


Copyright(c) SCSK Corporation, ALL Rights Reserved.

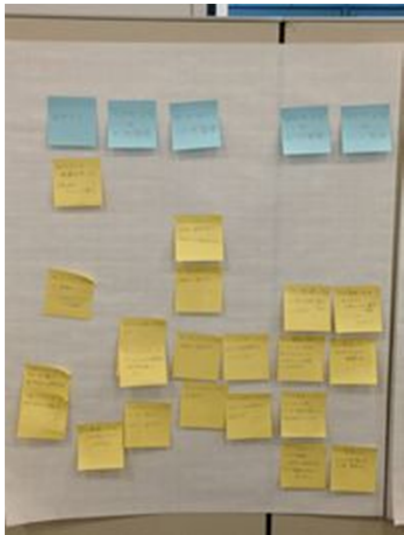
Redmineと アナログツールの併用

SCSK

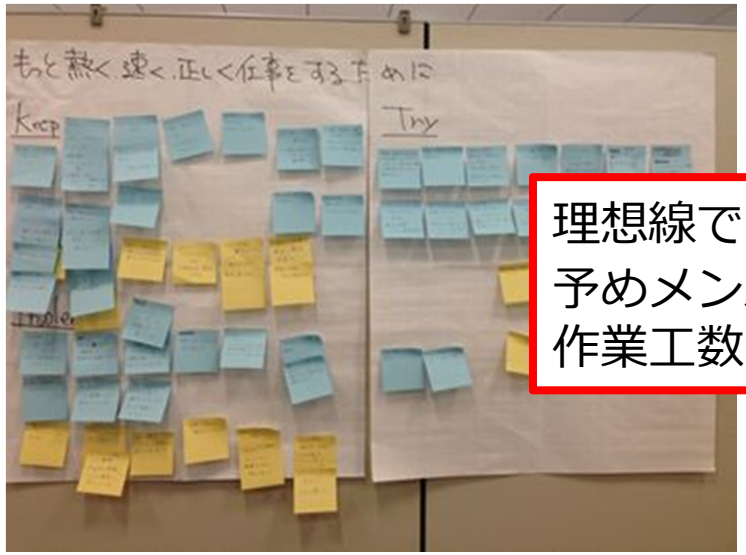
ベロシティ計測



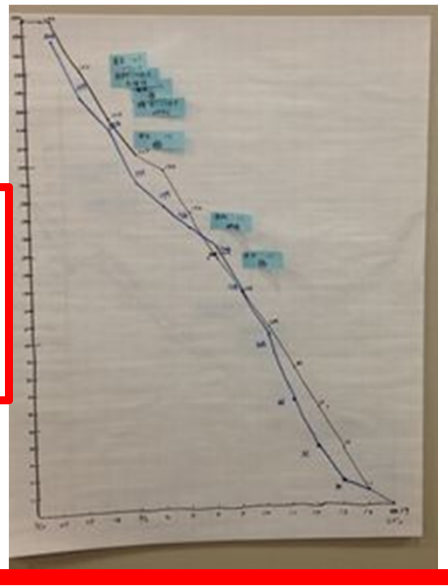
ユーザーストーリマッピング



レトロスペクティブ (プラクティス：KPT)



バーンダウンチャート



理想線ではなく**計画線**を導入。
予めメンバーの予定を記載し、
作業工数を実態に寄せる

Copyright(c) SCSK Corporation, ALL Rights Reserved.

それぞれのシーン別に実体験を交えてご説明

SCSK

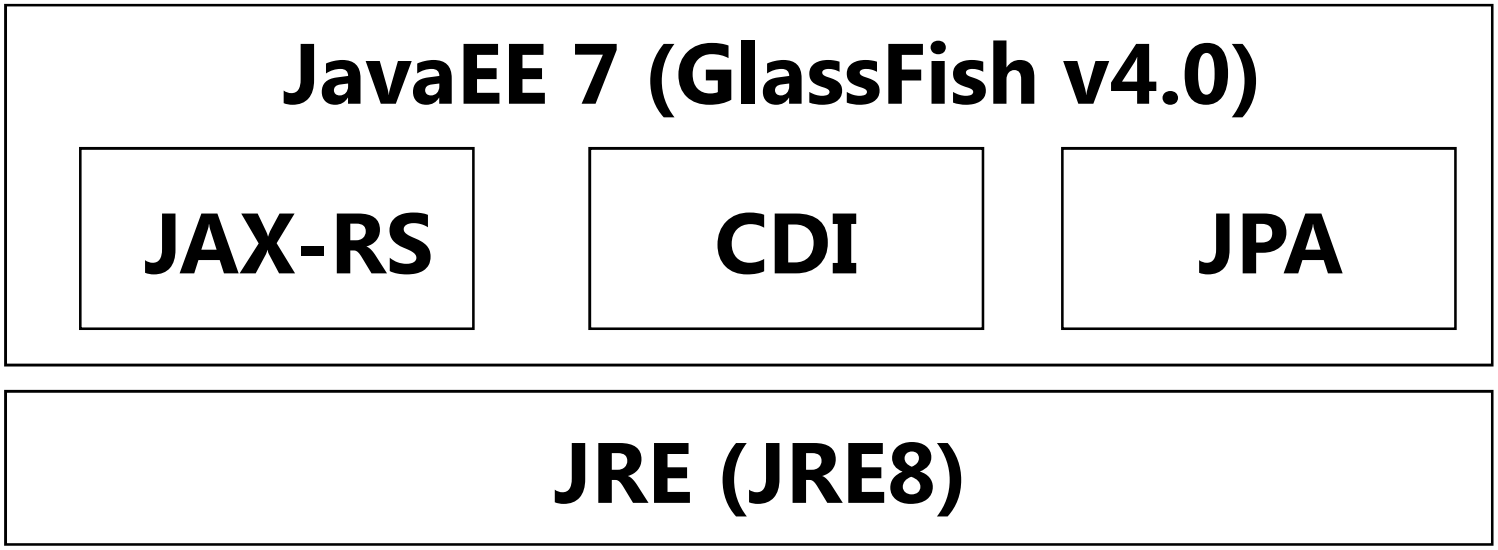
- いろんな開発プロセス・方法論ですすめる
  - ✓ アジャイル開発
  - ✓ **継続的インテグレーション (CI)**
  - ✓ **継続的デリバリー**
- 使ってみたい実装技術を影響の少ない部分で試す
  - ✓ **JavaEE 7 (2013年当時)**
  - ✓ NoSQL (Cassandra, MongoDB)
- 既存のOSSで代用できる部分は積極的に利用
  - ✓ ワークフロー

Copyright(c) SCSK Corporation, ALL Rights Reserved.

対象のアプリケーション

SCSK

- REST API を提供するWEBアプリケーション
- JavaEE が提供する基本仕様で実装



Copyright(c) SCSK Corporation, ALL Rights Reserved.

CI 環境として Jenkins を軸に構成

SCSK

プロジェクト管理・カンバン

Apache maven
Jenkins

自動テスト
自動デプロイ

vmware

仮想環境

ORACLE VM
VirtualBox

VAGRANT

個人開発環境

GitLab

git

ソースコード管理

Copyright(c) SCSK Corporation, ALL Rights Reserved.

CIの実現のために、テスト可能なアーキテクチャ設計が重要。

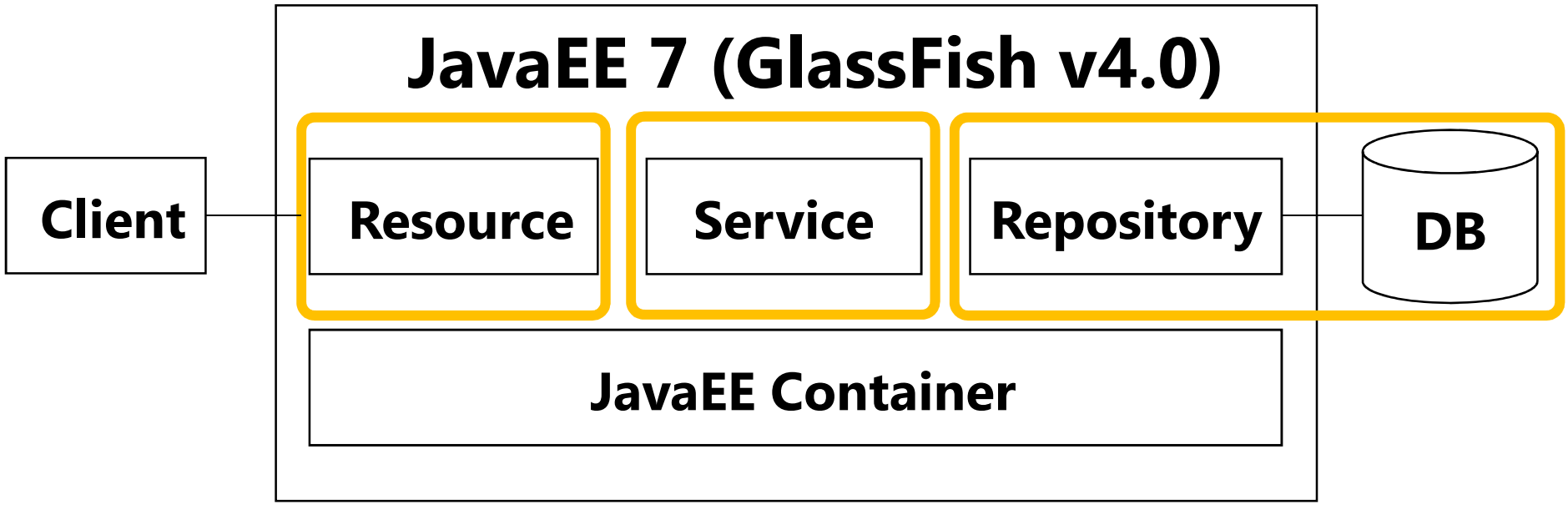
Copyright(c) SCSK Corporation, ALL Rights Reserved.

単体テスト設計 当初の計画

前提：Spring Frameworkの感覚で話していた。

- **Resource、Serviceクラスは JUnit + JMokit**
- **Repository クラスは DBUnitを使ってDBアクセスも検証**

と、なんとなく安易に単体テスト設計。



Copyright(c) SCSK Corporation, ALL Rights Reserved.

開発を開始していくと・・・

SCSK

- データソースをエミュレートしてくれるテスト用の仕組みがない。
  ⇒ データベースアクセスのテストができない
  ⇒ Arquillianというツールがあった。
- CI環境で動作しない。
  - ✓Arquillian の前のテストケースが使っている junit.jarの影響で ClassNotFound
- Arquillian は 実行速度が遅く、テストが増えてくるとCIサーバがダウンする。
- 挙句の果てにテストメソッドを日本語で書いて動かない問題まで。

Copyright(c) SCSK Corporation, ALL Rights Reserved.

メインの開発はストップ。
テスト環境の問題を
場当たり的に解決する日々

Copyright(c) SCSK Corporation, ALL Rights Reserved.

テスト計画に関する反省と考慮点

SCSK

- アプリケーションアーキテクチャを決定するときにはテストアプリケーションのアーキテクチャも合わせて設計する。
- 開発開始前に小さなサンプルアプリで開発からCI/CDまで検証を実施する。
- 失敗時の切り返し手順など、例外対応も重要。
- テスト系OSSツールは、開発系に比べるとプロダクト数、ネット上の情報も充実していないことを認識する。

Copyright(c) SCSK Corporation, ALL Rights Reserved.

プロジェクト管理の目標

SCSK

経験の少ないOSSや**技術・理論**を
試用してプロトタイプを実装し、

メインのターゲットである、
アプリケーション連携基盤の
アーキテクチャ及び
実現可能性の検証

を達成する。

Copyright(c) SCSK Corporation, ALL Rights Reserved.

本日のアジェンダ

1. OSS 活用の背景

2. OSS 活用で得たコト

3. まとめ

Copyright(c) SCSK Corporation, ALL Rights Reserved.

OSS を使用したプロジェクト管理のポイント

SCSK

- OSS を使ってPJで実現したい環境を整理し、自分の手で試す。
  - ✓WEBの情報を断片的に集めれば、実現可能なように錯覚するが、いろいろな組み合わせで問題が生じることもある。
  - ✓ベンダーが提供するような一式のものはなく、自分で組み合わせて開発するので、問題が生じやすい。
- 部分的な検証だけでなく、開発の流れに沿って小さなプロタイプでウォークスルーを実施する。
  - ✓すべてのリスクは拾えないが、すべて終わってみないとわからない問題はあらかじめ対処できる。
  - ✓テスト実行環境など、プロダクトで後回しにされやすいところなどを早めにつぶす。

Copyright(c) SCSK Corporation, ALL Rights Reserved.

それでも OSS を使う理由

SCSK

より**安く、うまく、早く**、開発するには**OSSは不可欠**。
新しい技術は OSS で実装される。ベンダーは年単位で遅れる。

課題や問題はあるものの、開発の開始前もしくは
なるべく早い段階で解決するようにする。

また同時にそのような**「リスク」をとってでも**
OSSを使う理由 があるのかを冷静に見極める。

OSSを使う場合には、例えサポートがあったとしても
問題が起きた時は **自分で解決するという覚悟** が必要。

さらにはその成果をコミュニティに還元することも含め、
大きな視点も持ちながら・・・。

Copyright(c) SCSK Corporation, ALL Rights Reserved.

OSSユーザのための勉強会 #11

Sep 25th 2015

OSS を使用した開発のプロジェクト管理

SCSK 株式会社

R&Dセンター 技術開発部 技術戦略課

岩本 健 (IWAMOTO Takeshi)

SCSK

夢ある未来を、共に創る。