平成２７年度文部科学省
成長分野等における中核的専門人材養成等の戦略的推進事業

富山県をモデルとした地方型グローバル
ＩＴエンジニアの育成

＜海外協働アプリ開発教材指導教員用教材＞

富山県をモデルとした地方型グローバルＩＴエンジニアの
育成カリキュラムの構築推進プロジェクト

学校法人浦山学園　富山情報ビジネス専門学校

目　次

1.課題説明

1.1 学習指導要領・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・ 1

1.2 受講にあたって・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・ 20

1.3 説明会と実習環境構築　課題説明・・・・・・・・・・・・・・・・・・・・・・ 53

1.4 スプリント実習　課題説明・・・・・・・・・・・・・・・・・・・・・・・・・ 60

1.5 スプリント（１～３）実習　課題説明・・・・・・・・・・・・・・・・・・・・ 65

1.6 成果発表　課題説明・・・・・・・・・・・・・・・・・・・・・・・・・・・・ 70

1.7 マスタースケジュール・・・・・・・・・・・・・・・・・・・・・・・・・・・ 73

2.スクラム実習

2.1 スクラム入門（基礎編）・・・・・・・・・・・・・・・・・・・・・・・・・・ 74

2.2 スクラム入門（実習編）・・・・・・・・・・・・・・・・・・・・・・・・・・ 104

3.開発関連

3.1Redmine(Backlog)の使用方法・・・・・・・・・・・・・・・・・・・・・・・・ 159

4.開発環境関連

4.1Redmine 環境構築手順書・・・・・・・・・・・・・・・・・・・・・・・・・・ 219

4.2Redmine 環境設定手順書・・・・・・・・・・・・・・・・・・・・・・・・・・ 235

5.指導員用

5.1 教員養成指導書 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・ 267

地方型グローバルITエンジニアの育成
ＰＢＬカリキュラム

学習指導要領

２０１５年度版

富山情報ビジネス専門学校編

目次

１．学習指導要領（全体）.....1
１．１．当カリキュラムの目標.....1
１．２．概要.....1
１．３．カリキュラム全体の構成.....4
１．４．学習指導要領(全体)の詳細.....5
２．学習指導要領（個別）.....7
２．１．カリキュラム説明会及び、開発手法の理解、受講環境の構築と各種ツール操作の習得、ユーザー要件の説明、スクラム入門（第１週～第２週：全２週間）.....7
２．２．スプリント１の演習（第３週～第６週：全４週間）.....9
２．３．スプリント２の演習（第７週～第１０週：全４週間）.....11
２．４．スプリント３の演習（第１１週～第１４週：全４週間）.....11
２．５．成果発表及び、最終評価（第１５週）.....11
◇補足資料.....13
補足 A.評価方法と指標について　【参考】.....13
補足 B. 海外連携とその準備について　【参考】.....16

１．学習指導要領（全体）

１．１．当カリキュラムの目標

IT 分野において日本と諸外国と言うような国家を跨いだコンピュータシステム開発(以下、オフショア開発)を実践するフィールドにおいて、中心となって活躍する事ができるグローバル IT エンジニアの育成を目指す為のカリキュラムです。　さらに従来までのオフショア開発のような国内で要件定義～設計までの作業を完了し、プログラム開発～単体テストと言った特に人的コストが高いフェーズのみを海外に依頼するのでは無く、設計作業からオフショア会社側担当者と共同で密接にコミュニケーションを取りながら開発を進めていく事ができる人材を育成することを目標とします。この開発形態は日本の地方に拠点を置く IT エンジニアに競争力を持たせる可能性が高くこれらの資質を持ったITエンジニアを特に地方型グローバルITエンジニアと呼ぶ事とする。地方型グローバル IT エンジニア(以下、IT エンジニア)とはオフショア開発を推進する環境において、異なる国もしくは国の出身者間の言葉や文化及び慣習の違いから発生する溝を埋めて円滑にシステム開発を進める為に必要とされるシステムエンジニアを指します。これらに加えて海外エンジニアと協働でシステム開発の設計～プログラム開発～納品までの作業を行うことができるシステムエンジニアを指します。 グローバル IT エンジニア育成として、英語による読み・書きを中心とした海外とのオフショア開発を体験する事からはじまり、更には ITSS アプリケーションエンジニアにおけるレベル２～３相当のスキル熟練度を習得する事を目指します。

１．２．概要

国内の地方中堅 IT 会社(以下、元請会社または元請)が受注したアンドロイドアプリケーションシステムの開発をアジアに位置する国外のソフトウェア開発会社をオフショア委託先(以下、オフショア側またはオフショア会社)パートナーとしてシステム開発を進める事を想定しています。現状は顧客が考えた要望をまとめたユーザー要件の作成を終了して、機能設計から当プロジェクトの運営に参画する形式でカリキュラムが開始されます。

開発手法としてアジャイル開発手法のスクラム開発フレームワークを使って開発を実施します。

その為、日本側学生は元請会社側として参加し、開発プロジェクト全体をコントロールし、サーバ及びデータベースの保守管理を実施し、オフショア側開発メンバーとコミュニケーションを取りながら設計・開発を協働で実施します。開発終了後、各自が作成したプログラムを統合して結合テスト（レビュー）を実施し、納品レベルまでソフトウェアの品質を管理します。

プロジェクトはスクラム開発フレームワークに従い、４週間＝１スプリント(イテレーション)として、全３スプリント(スプリント１～３)を実施します。(受講時間総数に合わせて２～４スプリントで実施します)

日本側学生は、元受会社社員として、上記の各フェーズにおいて主に以下の作業を管理します。

① ユーザー要件を基にして、開発対象となる要件を抽出し、その機能を設計・実装する為の作業を協働で行い、決められた期間内に正しく動作するシステムを構築出来る様に管理する。

② 元受側とオフショア会社側の話し合いで、開発対象の機能を選択し設計から開発、テストまでを協働で実施する。

③ 元請会社側の開発環境とオフショア会社側の開発環境の差異を調整して、オフショア会社側

から提出されるプログラムが問題無く動作する様に環境を整える。

④ 開発作業中の様々なトラブルに対し、状況を正しく把握して適切な解決策を英語にて行う事で信頼を獲得して、両社の協力関係が壊れない様に維持する。

⑤ 開発作業中に発生するデータベースの仕様変更やサーバとのインターフェース変更や仕様の変更を受入れ、管理する。

オフショア会社側学生は、主に下記の作業を実施します。

① 抽出された開発対象の要件から今回のスプリントで開発する対象を選定作業に参加し、元請側と協働で機能設計と開発及びテストを実施する。

② 毎日の作業において、朝会(ディリースクラム)を実施する。

③ 機能設計及び開発時に発生した問題点や疑問点を元請側と協働で解決する。

④ 各スプリント終了時に完成したプログラムを納品し、レビューに参加する。

⑤ 次のスプリントに向けて振り返り(レトロスペクティブ)に参加し、振り返り結果を元に次のスプリントで実装する要件・機能の決定に参加する。

⑥ データベースや開発環境などプログラム以外の変更が必要となった場合、元請側と打合せて変更に対応する。

当カリキュラムを実施する為に準備するものは、プロジェクト管理ツール(Redmine)導入用コンピュータ１台と、動作を検証する為のデータベース及び Web システムを導入する Web サーバ１台と、アンドロイド端末を数台、システムを開発・テストする為のコンピュータを各人１台必要となります。また必要に応じて開発したプログラムソースを管理する仕組みも利用します。

当カリキュラムは、元受側とオフショア側学生を合わせて５～８人で１つのグループを作って行うグループ学習が中心となります。　当該グループは IT エンジニアを想定しており、１つのグループ内訳は、スクラムマスター１名(講師が兼任可)、サブスクラムマスター１名（オフショア側、オフショア側のアドバイザーが兼任可)、プロダクトオーナー１名、開発メンバー３～５名で構成されます。プロジェクトのコントロールはスクラムマスターを中心として、チームメンバー全員で実施します。開発メンバーは、システムアーキテクト、プログラマー、テスター、データベース管理者などを複数兼任します。システムアーキテクトが開発環境及び Web サーバの運用管理を行い、スクラムマスター(元受側)とサブスクラムマスター(オフショア側)が窓口としてオフショア会社とのやり取りをサポートします。元請会社側の学生とオフショア会社側学生は、協働で発生した事象に対してチームとして結論を導き出し、解決策を講じます。

他方、元請会社に開発依頼している顧客(ユーザ)を教師が兼任します。教師は予め作成されているユーザー要件を理解し、プロダクトオーナーからの問合せや完成したプログラムのレビュー(デモンストレーション)に参加します。

開発終了時に学生は、実施した作業を振返ってプレゼンテーションを行います。

当該 PBL カリキュラム受講の前提として、付随する英語学習ｅ−ラーニング教材の受講完了が必要となります。

当教材は、元受会社（日本側）とオフショア会社（海外側）の２グループによる授業展開を想定していますが、海外のオフショア会社側の役割については、協力して頂ける英語圏の組織が無

い場合、国内のグループに置換えたり、オフショア側グループを設けない形式でも実施可能です。
この場合は、スクラム開発手法を学ぶための教材として利用する事になります。

□元請会社側学生がカリキュラム受講する前提スキルは下記の通りです。(上位ほど重要

- Java 言語による開発を経験している
- グループによるシステム開発を経験している
- アンドロイドアプリケーションの開発経験がある
- MySQL などデータベースを利用したシステム開発を経験している

□オフショア会社側学生の前提スキルは下記の通りです。(上位ほど重要)

- Java 言語による開発を経験している
- アンドロイドアプリケーションの開発経験がある
- グループによるプログラム開発経験がある
- MySQL などデータベースを利用したプログラム開発経験がある

※当カリキュラムにはオフショア会社側学生を評価する基準や方法は含まれていません。

１．３．カリキュラム全体の構成

当カリキュラムは、毎週６時間ｘ１５週＝９０時間(＋時間外作業)を標準構成としています。
第１週目から第１５週目までの学習内容は、下記の構成となります。

① 第１週～第２週	カリキュラム説明会及び、開発手法の理解、受講環境の構築と各種ツール操作の習得、ユーザー要件の説明、スクラム入門
② 第３週～第６週	スプリント１(開発環境の確認とベースプログラムの動作確認を含む)の実習。設計及び開発の実行と完成したプログラムの結合テストと納品。次のスプリントへの準備作業。教師が中心となり、スクラム開発手法を実践することにより、当手法を実践練習も含む。
③ 第７週～第１０週	スプリント２の実習。設計及び開発の実行と完成したプログラムの結合テストと納品。次のスプリントへの準備作業。
④ 第１１週～第１４週	スプリント３の実習。設計及び開発の実行と完成したプログラムの結合テストと最終納品。
⑤ 第１５週	プレゼンテーション準備と成果発表及び、最終評価

※実際にオフショア会社側担当を外部に委託(海外もしくは国内)する場合は、さまざまな考慮点が発生します。これらの例については、“補足 B.海外連携とその準備について”を参照。

１．４．学習指導要領(全体)の詳細

1.4.1.習得スキル、知識

- スクラム開発手法にそったシステム開発を実施する知識とスキル
- 直接コミュニケーションを取る事ができない作業者(オフショア会社)に対して、プロジェクト管理ツールや TV 会議ツール等を通して、英語を使ったコミュニケーションを取り、協働開発を進める為の順応性及び、知識とスキル
- 機能設計～プログラム開発までをオフショア側と協働で行う事を通して、交渉・調整する知識とスキル
- 国内の開発会社と異なる考え方や慣習に配慮した管理を実施する知識とスキル
- プロジェクト管理ツールの有効性を理解する知識と効果的な使い方ができるスキル
- アジャイル開発手法による開発を実施する為の知識とスキル

1.4.2.学習達成項目

- プロジェクトが破綻しない様に健全性を考慮しながら協働開発を行い、プロジェクトを完了まで導き、成果物を納品する為の技術を学ぶ
- 開発機能と作業の分担に関する交渉を成功させる
- 英語でどのような指示をすれば、相手が理解しやすいかを考える
- 機能設計を英語で体験し、設計内容を理解し承認する
- プロジェクト管理ツールを使う事による作業効率の良さを経験する
- それぞれの役割と立場を理解し、グループ作業を遂行する
- スクラム開発手法による開発を遂行する

1.4.3.KGI 項目

- オフショア会社側との協働開発を適切に実行できる
- オフショア会社側担当者とのコミュニケーションを英語で円滑に行える
- オフショア会社側との各種打合せを行える
- 指示された納期(３回)に成果物を納品できる
- プロジェクトの健全性を推量る事ができる
- プロジェクト管理ツールを正しく使う事ができる
- スクラム開発手法について理解する

1.4.4.教授指針

○前提知識

- 要件定義、設計、開発(プログラミング)及びテストの各工程について、その作業内容を理解してＰＢＬなどで開発を経験している
- Java 言語によるプログラミングの経験がある
- アンドロイドアプリケーション開発及び Web プログラミングの経験がある(無くても可)
- Web サーバ(PHP 言語使用)やデータベースサーバ構築の経験がある(無くても可)

○教授展開

・ スクラム開発手法によるシステム開発の一連の流れを確認しながら、教材テキストに加えてプロジェクト管理ツール(Redmine)を使用して授業を進める
・ 学生及び教師がメールを使用できて Redmine と連携出来れば更に良い
・ 実際に海外の学校と連携して実施する場合、先方への事前説明が必要となる

1.4.5.学習形式、学習教材及び、ツール

○学習形式及び、教材

・ 各教材テキストによる事前学習
・ グループ演習(元受側学生４～６名で１グループを構成)
・ プロジェクト管理ツール(Redmine)を使っての演習
・ Redmine を使った英語によるコミュニケーション(元請会社、オフショア会社間)
・ 日本語⇔英語の翻訳作業

○ ツール

・ プログラム開発ツール

Android Studio …アンドロイドアプリケーション用統合開発環境
Google API …GoogleMAP をアンドロイド上で表示する為の API

・ プロジェクト管理ツール

Redmine …業界標準のフリープロジェクト管理ツール(今回使用するメインツール)
遠隔地との共同作業及び、TV 会議などのコミュニケーションツール
(必要に応じて使用するツール)

・ 上記以外に利用するツール

Web サーバ及び、データベースサーバ
…アンドロイドアプリケーションと連携する Web サーバ

２．学習指導要領（個別）

２．１．カリキュラム説明会及び、開発手法の理解、受講環境の構築と各種ツール操作の習得、ユーザー要件の説明、スクラム入門（第１週～第２週：全２週間）

2.1.1.習得スキル、知識

- 当カリキュラムの目的と実施内容を理解する
- プロジェクトの背景と現状を理解する
- スクラム開発手法を理解する
- 受講環境の構築と各種ツールを利用するスキル
- ユーザー要件を理解する

2.1.2.学習達成項目

- 当カリキュラムの目的と実施内容を理解する
- プロジェクトの背景と現状を理解する
- スクラム開発手法をシミュレーションを通して理解する
- プログラム開発環境を構築し、使い方を習得する
- プロジェクト管理ツールの使い方を習得する
- ユーザー要件を理解して開発する対象をイメージする

2.1.3.KGI 項目

- 当カリキュラムの目的と実施内容を理解
- 当プロジェクトの難しさを知る
- スクラム開発手法による開発方法を理解
- プログラム開発環境を正確に構築する
- プログラム開発環境を正しく使う
- プロジェクト管理ツールを正しく使う
- プロジェクト管理ツールの利点と欠点を理解
- ユーザー要件を正確に理解して開発対象をイメージできる

2.1.4.教授指針

〇前提知識

- 要件定義、設計、開発(プログラミング)及びテストの各工程について、その作業内容を理解してＰＢＬなどで開発を経験している
- Java 言語による Web プログラミングの経験がある
- Web サーバなどサーバ構築の経験がある

〇教授展開

- 当システム開発の目的を説明し、スクラム開発手法を用いたシステム開発の一連の流れを説明しながら、教材テキストに加えてプロジェクト管理ツール(Redmine)の使い方を実習する
- スクラムシミュレーションを通して、スクラム開発手法を体験学習する

・ ユーザー要件を確認してこれから開発するシステム全体像をイメージ出来る様に説明する
・ システム開発に必要なプログラム開発環境を構築手順書に従って、学生の PC 上に構築し、使い方の基礎を実習する
・ 構築手順書に従って Web サーバ及びデータベースサーバを構築し、使い方の基礎を実習する
・ 構築手順書に従ってプロジェクト管理ツール(Redmine)を共用サーバに導入し環境を整える

2.1.5.学習形式、学習教材及び、ツール

○学習形式及び、教材

・ 各教材テキストによる事前学習（スクール形式）
・ グループ演習(元受側学生４～６名で１グループを構成)
・ グループ内の役割分担を決定
・ プロジェクト管理ツールの使用環境構築と動作確認による使い方の学習
・ スクラム入門としてのスクラムシミュレーション実習
・ 使用する教材

演習概要説明「受講にあたって」
ユーザー要望「User Requirements List」
プログラム開発環境構築と使い方の説明　「アンドロイド開発環境構築手順書」
Web サーバ環境構築と使い方の説明　「テスト用サーバ環境構築手順書」
プロジェクト管理ツールの使い方説明「Redmine 使用方法（Backlogs）」
プロジェクト管理ツールの構築　「Redmine(Backlogs)サーバ構築手順書」
スクラム入門 「Scrum Simulation1 Introduction」、「Scrum Simulation2 Practice」

○ツール

・ 下記のプロジェクト管理ツール

Redmine
Skype（TV 会議&チャットシステム）

・ 下記のプログラム開発環境

Android Studio
Web サーバ及びデータベースサーバ（テスト用及び検証用）

2.1.6.評価基準

・ 環境構築手順に従ってプロジェクト管理ツールの導入ができたか（講師による事前準備）
・ プロジェクト管理ツールを理解して使えるか
・ 環境構築手順に従ってプログラム開発環境の導入ができたか
・ プログラム開発環境を理解して使えるか
・ スクラム開発手法をシミュレーションを通して理解できたか
・ その他質疑応答等により演習内容の理解度を確認する

２．２．スプリント１の演習（第３週～第６週：全４週間）

2.2.1.習得スキル、知識

- スプリントの実施内容と手順を理解し、開発環境とベースプログラム及び Web サーバが正しく動作することを検証する知識
- スクラム開発手法によるシステム開発スキル
- 直接コミュニケーションを取る事ができない作業者(オフショア会社)に対して、英語を使って、協働開発する為の順応性及び、知識とスキル
- 国内の開発会社と異なる考え方や慣習に配慮した管理を実施する知識とスキル
- プロジェクト内での役割を理解し、率先して作業を実施する知識とスキル
- Redmine の有効性を理解する知識と効果的な使い方ができるスキル

2.2.2.学習達成項目

- スプリントを実施する上での開発環境とベースプログラムの詳細を理解する
- スクラム開発手法を使ったシステム開発を経験する
- 英語でどのように表現すれば、相手が理解しやすいかを考える
- オフショア開発側担当者とコミュニケーションを上手に取ることができる
- 自分の役割を理解し、率先して作業を実施することができる
- Redmine を使ったプロジェクト管理の作業効率の良さを経験する

2.2.3.KGI 項目

- 使用する開発環境とベースプログラムがどうすれば正しく動作するかを理解する
- スクラム開発手法による開発で何を報告しなければならないかを理解する
- オフショア会社側と協働でプログラムを開発し納品できる
- スクラム開発手法を使ったシステム開発を説明できる
- オフショア会社側にこちらの意志を英語で伝える事ができる
- オフショア会社側担当者とのコミュニケーションを英語で円滑に行える
- 役割に応じた作業を正しく行える
- Redmine を正しく使う事ができる
- プロジェクトの健全性を推量る事ができ、改善策を提示できる

2.2.4.教授指針

○前提知識

- 要件定義、設計、開発(プログラミング)及びテストの各工程について、その作業内容を理解してＰＢＬなどで開発を経験している
- Java 言語によるプログラミングの経験がある
- プロジェクト管理ツールの使い方を理解している
- スクラム開発手法を学習している
- Redmine の使い方を理解している

○教授展開

- スクラム開発手法のシステム開発の流れ(元受会社側及びオフショア会社側の両方)を再確認しながら、教材テキストに加えて Redmine を使用して授業を進める
- 開発環境とベースプログラムが正しく動作することをプログラム開発を通して検証する
- プロダクトバックログ→スプリントバックログ→担当者割当て→デイリースクラム→進捗更新→納品→ふりかえりまでの行程を Redmine 上で実際に運用する
- プログラム開発中にシステムアーキテクトに依頼する内容(DB テーブル及び PHP スクリプトの変更など)を実習を通して理解する
- オフショア会社側がどのような行動を取るかを予測して作業を進めるように常に考える
- 開発先と直接会話できない不自由さと異なる言語による不自由さの中でどうすれば相手に正確に伝えることが出来るかを常に考えて作業を行う事を求める
- 実際に外国人とチームを組んで作業を実施する場合、相手国とその文化・習慣を尊重し、自分自身を過小評価せずに、自分たちが日本および自分が所属する教育機関を代表しているという意識と自信を持って接する態度が重要である事を説明する
- 各スプリントにおけるユーザー要件(プロダクトバックログ)をスプリントバックログにする比率は下記を基本とする

  スプリント１ ＝ ユーザー要件の２０％
  スプリント２ ＝ ユーザー要件の５０％
  スプリント３ ＝ ユーザー要件の３０％（前スプリント見直し分も含む）

2.2.5.学習形式、学習教材及び、ツール

○学習形式及び、教材

- 各教材テキストによる事前学習
- グループ演習(元請側及びオフショア側合わせて５～８名で１グループを構成)
- Redmine を使っての演習
- e-Mail を使ったコミュニケーション(元請、オフショア間)
- 日本語⇔英語の翻訳作業
- 教師による学生個人毎の理解度評価の実施

○ツール

Redmine　…世界標準でフリーのプロジェクト管理ツール
アンドロイドアプリケーション開発環境
　…Android Studio アプリケーション開発環境ベースプログラム一式
Web サーバ　…Apache Web サーバと PHP 及び MySQL サーバが導入された
　サーバ環境（個々のテスト環境及び、動作検証用環境）

2.2.6.評価基準

- 当スプリント終了時点で学生毎の理解度を評価する指標を使って評価を実施する。(評価と指標については、“補足 A.評価方法と指標について”を参照)
- 開発環境とベースプログラム及び Web サーバが正しく動作することを検証できる
- スクラム開発手法によるプログラム開発の手順を理解し、説明できるか
- どのように当該アプリケーションが動作するかを説明できるか
- スクラム開発手法によるプログラム開発で Redmine をどのように利用するかを説明できる

２．３．スプリント２の演習（第７週～第１０週：全４週間）

- スプリント２の演習の内容については、上記“２．２．スプリント１の演習”と同様になります。

２．４．スプリント３の演習（第１１週～第１４週：全４週間）

- スプリント３の演習の内容については、上記“２．２．スプリント１の演習”と同様になります。

２．５．成果発表及び、最終評価（第１５週）

2.5.1.習得スキル、知識

- 当カリキュラムで学んだ事を振返って整理する
- スクラム開発手法の長所と短所を知る
- オフショア会社と協働によるシステム開発の難しさや特異性を知る

2.5.2.学習達成項目

- 当カリキュラムで学んだ事を整理して発表する
- スクラム開発手法と他の開発手法との違いを理解し、どのような長所と短所があるかを理解する
- 元受会社チームとして作業を行い良かった点、悪かった点を理解する

2.5.3.KGI 項目

- 当カリキュラムの目的と実施内容を理解
- 当カリキュラムで学んだ事を発表できる
- 教師や他の生徒からの意見を取り入れる

2.5.4.教授指針

○前提知識

- 当カリキュラムの３回のスプリントを実習し、一通り終えている
- アジャイル開発手法にて開発されたアプリケーションと Web サーバが動作する状態で手元にあること

○教授展開

- 教材テキストに沿ってプレゼンテーション資料をまとめて、個々の学生が発表を行う
- 発表内容に対して教師及び学生が意見を述べ合う

2.5.5.学習形式、学習教材及び、ツール

○学習形式及び、教材

- 各教材テキストによる事前学習（スクール形式)
- グループ演習(元請会社側学生４～５名で１グループを構成)
- 学生個別の成果発表プレゼンテーション
- 使用する教材

  演習概要説明「成果発表 課題説明」

2.5.6.評価基準

- プロジェクトを完了まで導けたか
- 開発中に発生した問題点を解決して役割を遂行できたか
- スプリントを重ねる毎に成果物の完成度が高くなったか
- 当カリキュラムを通して得た知識やスキルを整理できているか
- スクラム開発手法による開発を理解し整理できているか
- プロジェクト管理ツールを適切に活用できたか

当評価基準は、３回のスプリントでそれぞれ実施してきた学生毎の理解度を一定の基準に照らし合わせて評価したものを総合して最終的な評価とします。(評価基準の詳細については、“補足 A. 評価方法と指標について”を参照）.

以　上

◇補足資料

補足 A.評価方法と指標について 【参考】

学生に対する評価は教師が行い、全３スプリントそれぞれで個々のスプリントにおいて、それぞれの学生を評価します。評価の為の情報は、各学生の開発プロジェクトへの参加度合いとオフショア会社側とのコミュニケーション及び、担当している作業内容の理解度と積極性を基本とします。これらの情報は、スプリント中の学生の言動や、Redmine 上の受答えと報告実績、スプリント終了時のアンケート内容及び、教師による質疑応答によって収集します。

学生の理解度評価と指標

授業で実施されるプロジェクトは、スプリント単位に開発対象の選出(バックログ)と機能設計、開発、振返り、納品及び、次スプリント準備のフェーズに分かれており、それぞれのフェーズにおける役割の理解と実施状況を判断します。また評価者は教師のみとし、全３スプリントの合計を最終評価とします。

○５つの評価項目：

1. **目的の理解度** …割当られた役割に要求されている事は何でどのような結果を求められているかについて正確に把握できているかどうか。
2. **作業の実行度** …指示された役割の目的達成に向けて正確に作業を実行できたかどうか。また、自主的が出来る作業を探して実行できたかどうか。
3. **問題解決力** …質問や発生した問題に対して、プロジェクトがより良い方向に向かう事を考慮して判断し、的確に行動できたかどうか。
4. **コミュニケーション力**…元請側で学生同士及び、オフショア会社側とのコミュニケーションが円滑に正しく行われているかどうか。
5. **ツール理解度** …Redmine の使い方を正確に理解し、効率よく使用できるかどうか。自分が必要とするその他ツールについても適切に利用できたかどうか。

○６つの指標：

5(最も高い) …評価項目を理解して実施し、目的通りの結果を得る事ができる。
4 …評価項目を理解して実施し、目的に対して８０％の結果を得る事ができる。
3 …評価項目を理解して実施し、単独で５０％の結果を得る事ができる。
2 …評価項目を理解しているが、ほとんどが他者からの指示で実施できる。
1 …評価項目をほとんど理解しないで、プロジェクトに参加しているのみ。
0(最も低い) …全く評価項目を理解できず、進展も見られず成果物を提出できない。

○得点について

学生が得る得点は、上記５つの評価項目をそれぞれ６つの指標(０～５)に照らし合わせて評価するが、評価対象のスプリント(１～３)によって満点が以下の様に異なる。最終的な得点は３つのスプリントの合計によって求められる。これはスプリントを繰り返すことで理解度が上がり、より高度な作業を実施できる様になる事を想定してスプリント毎の得点重み付けを変化させている。

スプリント１　・・・　２５点満点　（１指標ポイント＝１点）
スプリント２　・・・　２５点満点　（１指標ポイント＝１点）
スプリント３　・・・　５０点満点　（１指標ポイント＝２点）

(例　スプリント１の評価合計＝１５ポイント　x　１点　＝１５点
スプリント２の評価合計＝２０ポイント　x　１点　＝２０点
スプリント３の評価合計＝２２ポイント　x　２点　＝４４点
合計　＝７９点

○学生の評価と理解度について

最終的な学生一人一人の評価は、各スプリントにて取得した得点の合計点を積み上げて、１００点満点で基準に合わせて判定します。学生が当カリキュラムの学習目標を達成していると判断する基準は、６０点(１００点満点中) 以上とします。

スプリント終了時 アンケート （サンプル）　　作成日付： 20 . .

氏名：

役割：

以下の質問に答えなさい。またそれぞれの質問内容について自分の取組んだ状況を５段階の自己評価（５：良い ～ １：悪い）をしなさい。

【質問１】このスプリントにおけるあなたの役割は何でしたか？どのような作業を行いましたか？

（回答１）

自己評価＝

【質問２】あなたは、このスプリントでどのような問題にぶつかりましたか？ またその問題をどのようにして解決しましたか？

（回答２）

自己評価＝

【質問３】当スプリントにおいて、Redmine を使って運用しました。どのように Redmine を活用しましたか？ またどの機能が便利でどの機能が必要ないと思いましたか？ その理由も述べなさい。

（回答３）

自己評価＝

【質問４】国内側メンバーと、活発にコミュニケーションを取れましたか？ またコミュニケーションを取るためにどのような工夫をしましたか？ 詳しく書いて下さい。

（回答４）

自己評価＝

【質問５】オフショア側メンバーと、活発にコミュニケーションを取れましたか？ またコミュニケーションを取るためにどのような工夫をしましたか？ 詳しく書いて下さい。

（回答５）

自己評価＝

補足 B. 海外連携とその準備について 【参考】

当カリキュラムは、日本で受講する学生がグローバル IT エンジニアとしての資質を身に付ける為の教材となっており、元受会社側とオフショア会社側の異なる立場のチームが実習を通して様々な技術を身につけることを想定しています。もし、当該カリキュラムを実行する際に海外の学校(もしくは日本語を母国語としない団体)の協力を得られる場合、オフショア会社側チームを海外の学校に依頼できれば、より実践的な内容となりグローバル IT エンジニア育成教材として効果が高くなります。

海外の学校と連携する場合は、当カリキュラムの主旨や、先方に依頼する内容、対象となる学生のスキルレベルなどを通知する必要があります。またカリキュラムを実際に実施する前に、１ヶ月程度の実証講座を企画し、日本側及び海外の学校側から数名の学生を選出して頂き、試験的に講座を実施して事前に問題点や調整事項などを洗い出しておくことを推奨します。

当カリキュラムの作成と検証の為に、フィリピン マカティ大学の協力を得て、実証講座(約３ヶ月)を実施しました。その際に使用した説明資料(日本語、英語)や実証結果も当カリキュラムの補足資料として添付されていますので、必要に応じてご利用下さい。

このような実証講座を実施する場合、国内・国外両校のトップレベルの合意が最も重要となり、これらの間を取り持つコーディネータの存在が非常に大きいと感じています。トップレベルの合意がとれた場合、文化や商習慣にもよりますが、２～３ヶ月の準備期間を取ることができれば、海外と連携した形態でのカリキュラム実施が可能になると考えられます。

当カリキュラムを実習するための環境とスキルについて

○対象となる学生のスキル

(1)複数人数で実施する一般的なシステム開発の経験がある

(2)Java 言語によるアンドロイドアプリケーションの開発経験がある

(3)MySQL などデータベースを利用したプログラム開発経験がある

○実習環境

(1)常時インターネットへの接続が可能な環境(ソフトウェアのダウンロード含む)

(2)日本へのインタネットアクセスが可能な環境

(3)学生一人ひとりが開発用 PC を利用でき、ある程度の期間専有できる環境

(4)学生一人ひとりが e-mail を送受信できる環境

(5)Skype など TV 会議及びチャットシステムを利用できる環境

外部の協力を得て当カリキュラムを実施する場合のサンプルスケジュール

下記は外部(海外もしくは国内)の委託先にオフショア側実施を依頼し、当カリキュラム受講者は元請け側として作業を行うことを前提としています。

	No	項目	実施1-2ヶ月前	第１月目 W1-W4	第２月目 W5-W8	第３月目 W9-W12	第４月目 W13-W15
オフショア側	1	協力体制構築と作業依頼					
	2	作業内容の説明会					
	3	環境構築					
	4	スプリント１実施					
	5	スプリント２実施					
	6	スプリント３実施					
	7	成果発表会実施					
元請け側	8	カリキュラム説明会と環境構築					
	9	スプリント１実施					
	10	スプリント２実施					
	11	スプリント３実施					
	12	成果発表会実施					

※オフショア会社側の作業を外部に委託する場合、実施して頂ける外部機関を選定し、協力体制の確立と作業内容を説明し対象者の選定を実施して頂く。次に協力対象者に対して作業内容の説明会を実施します。ここまでの作業を授業開始の１ヶ月前までに完了する必要があります。

※受講する学生の心構えとして、外国のチームと接する場合、自分たちが日本および所属する教育機関の代表との意識を持ち、自分自身を過小評価せずに自信に満ちた態度で接することが特に重要になります。また、相手国の文化と習慣を尊重し、相手に敬意を払って対等な立場で接することが重要であることを必ず学生に通知する必要があります。

※授業が開始され当カリキュラム受講対象者へのカリキュラム説明会と環境構築が完了して外部委託者側での環境構築とその内容確認を実施し、スプリント１作業を開始する準備を整えて頂きます。

※スプリント１～スプリント３実施時は、元請会社側とオフショア会社側が連携しながら授業を進めます。※最後の成果発表会は、可能であればオフショア会社側作業者と元請会社側作業者を集合させて合同で実施すると、さらに高い学習効果が望めます。

以　上

1．受講の前提について

◯学生が、当カリキュラムの受講に当って下記の前提条件が必須となります。付属のサンプルソースを使用する場合、下記の前提知識が無い場合、まずはこれらの学習を受講した後に当カリキュラムを受講して下さい。また、付属のサンプルソース以外の開発言語や開発対象のプログラムを別途作成して実施する場合は、受講条件はその開発対象の言語及びプログラムに依存します。

- ・当カリキュラムに付随するe-Learning教材の受講
- ・Java言語によるアプリケーションの開発経験
- ・SQLによるデータベース操作の経験
- ・Microsoft Internet Explorerなど一般的なWebブラウザの操作
- ・受講する学生間でチャット(SkypeやLINEなど)を行うための環境と操作経験

◯学生が、当カリキュラム受講前に下記の予備知識があれば、受講がスムーズになります。

- ・MySQLのデータベース操作経験
- ・PHP言語による開発経験
- ・チームによるアプリケーションの開発経験
- ・Androidのアプリケーション開発経験
- ・Webサーバの構築経験

◯当カリキュラムを担当する講師に必要な前提知識は下記の通りです。

【必須知識】

- ・Java言語によるアプリケーションの開発経験

・SQLによるデータベース操作の経験
・Microsoft Internet Explorerなど一般的なWebブラウザの操作
・MySQLのデータベース操作経験
・PHP言語による開発経験
・チームによるアプリケーションの開発経験
・Androidのアプリケーション開発経験
・Webサーバの構築経験

【あれば尚良い知識】
・Linux OS（CentOSなど）のインストール経験
・Redmineによるプロジェクト管理経験

2．当教材の使い方と目的

◯当教材は、グローバルIT人材育成の為に、チームによるプログラム開発力、スクラム開発手法の習得、英語によるIT関連のコミュニケーション力、Java言語によるAndoroidアプリケーションの開発を身につけることを想定しています。

◯受講者のレベルや身につけさせたい技術などの目的に応じて、内容をアレンジすることで、上記の目的の中からいくつかを選択して受講者に学習させることも出来ます。

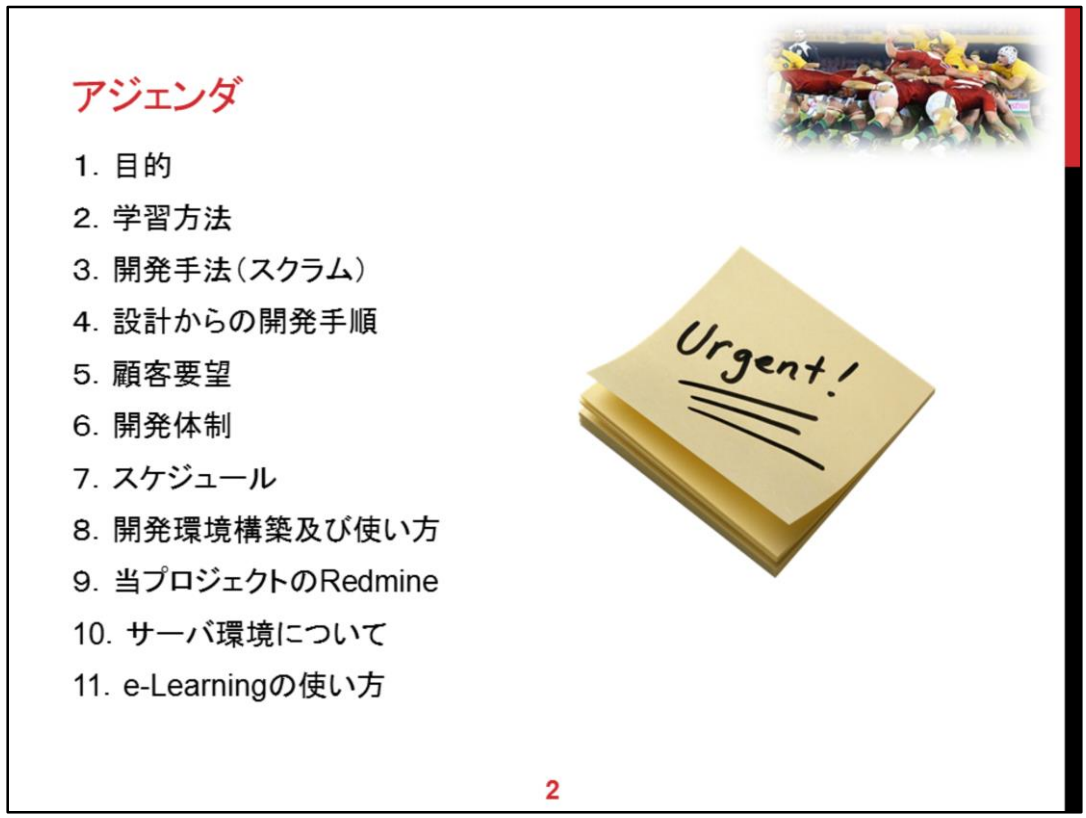
3．講師側の事前準備について

◯当教材を実施するにあたって、Redmineサーバの構築と必要なBacklogsアドインの導入を事前に実施しておく必要があります。また、クライアント側開発環境であるAndroid Studioを利用する際は、講義開始直前に講師側で一度インストールを実施し、最新版のAndoroid Studioでサンプルソースが正しく実行される事を確認しておく必要が有ります。また場合によってはAndoroid Studioの自動更新を行わせないように設定し、受講者にも更新を実施させないように指示する必要がある場合があります。

◯もし実機のAndoroid端末をグローバルIPを利用して利用する場合は、インターネット上に公開されたWebサーバ（PHP＋MySQL）の環境を事前に構築し、サンプルソースを導入して動作確認を行っておく必要があります。受講するチームの数に応じてWebサーバのアクセス先URLを予め定義しておきます。

◯構築されたRedmineサーバ上に、この講座で使用するすべてのドキュメント及びサンプルソースを予めアップロードしておきます。

◯当カリキュラムで使用しているe-Learningシステムについては、今回株式会社デジタル・ナレッジ様より当カリキュラム用に使用許可を得ています。同様なe-Learningシステムを利用される場合は、当e-Learningシステム管理会社様までお問合せ下さい。



1．講義の順番について

◯アジェンダ上は、e-Learningの使い方が最後に来ているが、前もってe-Learningを受講しておいた方が当カリキュラムをスムーズに実施できる。この為、１１．e-Learningの使い方から説明を開始しても良いです。

◯また3．開発手法（スクラム）に関する学習は、事前学習として最も重要なものに位置づけられます。こちらも事前授業としてe-Learningと同様に先に説明を開始しても良いです。

1. 目的
1.1 プロジェクトの目的

- 当教材は学生が英語によるコミュニケーションをより高度に発展させグローバルなITエンジニアとしての資質を備える事を目的とする。
- 特に企画提案作業から海外エンジニアと共同でシステム開発を実施することにより、企画提案力及びリーダシップと協調性を身につける。
- スクラム開発技法を理解し、システム開発を開始～完了まで実践を通して体系的に学ぶ。
- 当教材を使用することにより、学生がどのように成長したかを客観的に判断できる。

3

◯グローバルITエンジニアとは、国内だけでなく海外のエンジニアとも協働でシステム開発を行える人材を指します。協同作業とは、ユーザからの要望を理解しどのようにアプリケーションに実装するかを提案し、話し合いで決めることができて、開発作業中に発生するさまざまな課題をチームとして解決し、開発されたプログラムを厳密にレビューして、ユーザの為にリリースする一連の作業を日本人と海外エンジニアの連携の元で実施することを指しています。

◯受講者が、おそらく初めて接する“スクラム開発技法”を用いることにより、上記の協同作業をすばやく円滑に実施できるようになる事を説明します。また、成功の鍵として緊密なメンバー間のコミュニケーションが欠かせないことも合わせて説明します。スクラム開発においては、発生した課題や問題点を一人で抱え込まないで、メンバー全員で解決する為に努力しなければならない事も追加で説明します。

1. 目的
1.2 当講座の目的

- アンドロイドアプリケーション開発を通して、日本の学生達が海外の学生達と、どのように協調してシステム開発を遂行できるかを学ぶ。
- 当講座を受講した学生の態度変容を促します。
- 開発は、複数のチームが同時に同じユーザ要望を基に、それぞれのチームで機能を発展させ、独自性を持たせます。
- 教材として準備されたユーザ要望を実現する為に、日本と海外の学生が共同で設計作業～製造～納品を複数回実施できる事を学ぶ。
- アジャイル開発手法(Scrumフレームワークを使用)によるシステム開発をどの程度理解し、実行できる事を学ぶ。
- Redmineによるプロジェクト管理及び、システム開発を実施する方法を学ぶ。

4

◯スクラム開発技法を学習する為のサンプルとして、Andoroid端末で動作するアプリケーションを開発する事を説明します。このアプリケーションは、Google MAP APIとGPSの機能を駆使して動作するものです。開発プロジェクトの運用管理、及び、記録を残すためにRedmineというプロジェクト管理システムを使用することを説明します。また、すべてのメンバーが直接会話できない環境の場合は、TV会議システムは、チャットシステムを追加のコミュニケーション手段として使用する事を説明します。

◯海外のエンジニアとの協働開発を想定している為、会話及びRedmine上のやりとりは、原則英語にて行うことを説明します。英語によるコミュニケーションに不安がある場合は、事前学習のe-Learningを必要に応じて何回でも見直す事を推奨して下さい。

1. 目的
1.3 当講座の開発対象

- 旅行に来た外国人観光客が便利に使えるアンドロイドアプリケーションを開発する。現在地点の地図情報を利用した観光スポット紹介アプリケーションとする。(GoogleマップAPIを利用する)
- ユーザが作成した要望事項一覧に従って、日本の学生が仕様を決定した後、日本の学生と海外の学生が、開発から納品までを共同で実施する。
- 今回使用するベースシステム(アンドロイド及びWebサーバ)は、アーキテクトによって構築されている。また実際に動作するサンプル・アプリケーションのソースプログラムも提供される。これらを拡張して目的のシステムを構築する。
- 日本と海外学生との混成チームが複数同時に開発作業を実施する。但し、チーム間では出来るだけ情報交換出来ない状況を前提とする。
- サンプルプログラムは、Java言語(クライアント側)及びPHP言語(サーバ側)で記述されているが、チームの状況によっては他の言語を使用する事を考慮しても良い。但し、この場合、開発チームメンバー全員で決めて周知し、開始して下さい。

5

◯対象となるアプリケーションは、Andoroid端末上で動作し、Google MAP APIを使用した地図上に、GPSより取得した自分を位置を表示しながら動作する事を説明します。また、機能はサンプルソースとして一部実装済みの状態で提供されており、提供済みの各画面は次のスライドにて説明します。一般の不特定多数の人を利用者として想定しているアプリケーションなので、見栄えを重視することも忘れないで下さい。

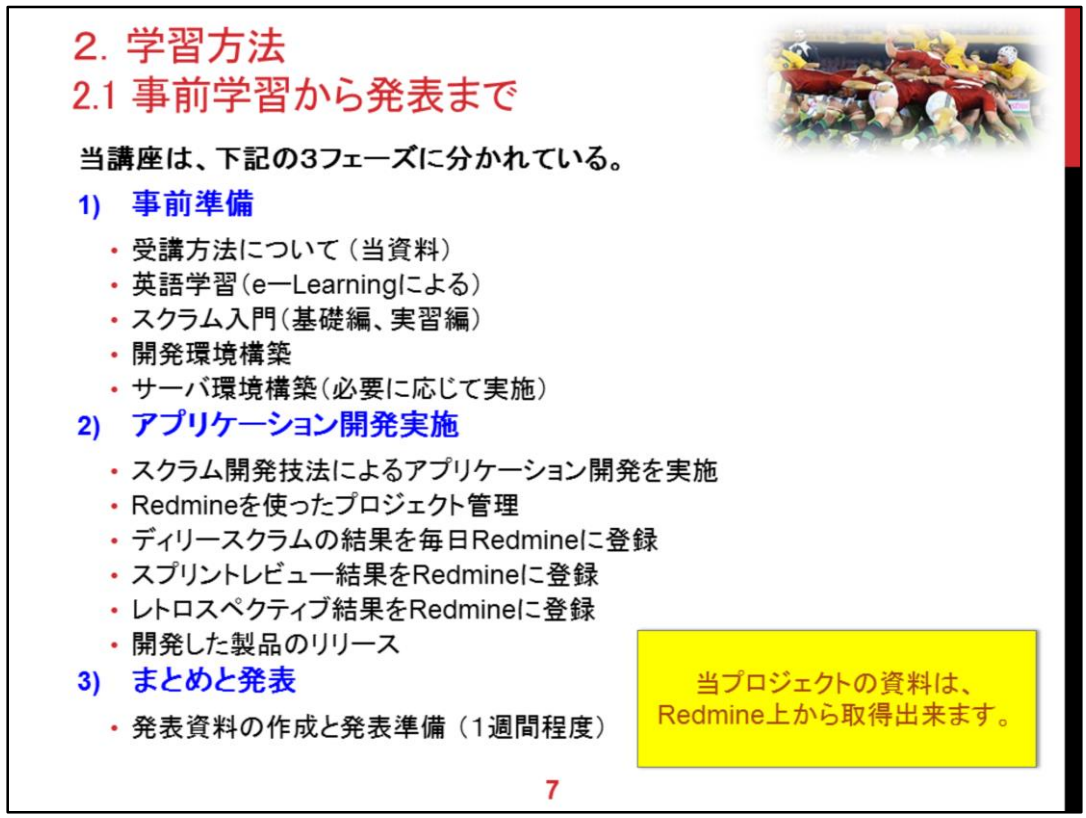
◯顧客要望事項一覧として提供されているドキュメントに従って、実装する機能を各チームで決めることになりますが、記載されている機能の置き換えや拡張、新機能の追加などを行うことも許可します。最終的にどうのような機能を開発側が提案し実装するかについては、ユーザと開発チーム間の打合せで決めていくことになります。現在の要望事項にとらわれ過ぎないで、自由な発想で魅力的なアプリケーションを開発させるように指示して下さい。

◯サンプルソースを利用する場合、Andoroidアプリケーションの開発として、Java言語及び、Webサーバへのアクセスに関する知識が必要となります。データベースサーバ側の開発として、PHP言語によるデータベースのアクセス方法及び、SQLによるデータベース操作に関する知識が必要になります。

◯サンプルソースとして既に実装されている画面の一覧です。
・アプリケーションのアイコンは、デフォルトの味も素っ気も無いものが使用されています。
・Andoroid端末上ではじめてこのアプリケーションを使用する場合、利用者のユーザIDとパスワード登録を行う画面を表示します。この画面は新規端末での実行時に1回だけ表示されます。
・2回目以降の起動であれば、このメインメニューからアプリケーションが開始されます。メインメニューにはこのアプリケーションのバージョン情報を表示する機能を備えています。
・Display Map画面では、GPSより取得した現在位置を中心とした地図表示を行います。地図上にはデータベースサーバから取得した付近の登録地点のマーカーが表示されます。
・マーカーをクリックすると、そのマーカーが指している地点(お店)の詳細情報を表示します。
・画面上で検索文字を入力して検索を実行することで、対象となるデータを表示することが出来ます。
・地図上でマーカーの無い場所で、長押しするとその地点の情報を新規登録する画面が表示され、付帯情報を入力して登録を行うことでデータベースに情報を追加することが出来ます。
・データベース上に追加されているデータを元にした各種ランキングを表示します。

◯各チームで、現在実行されている機能の拡張や置換え、新機能の追加を行って下さい。まずはサンプルソースを解析して、Andoroidアプリケーションの

開発に関するコーディングの作法や、Google MAP APIの実装方法を良く研究して下さい。またWebサーバ上のデータベースにアクセスするためのPHPスクリプトの書き方とSQLの実行方法をWebサーバ側のソースを確認して記述方法を調査して下さい。またJava言語とPHPスクリプト間でのパラメータの受渡しについても、慎重に調査し、理解しておいて下さい。



0. 資料について

◯このカリキュラムには、非常に多くのドキュメントが含まれています。これらすべてのドキュメントはRedmine上にアップロードされており、いつでも取得する事が出来ます。

1. 事前準備

◯事前準備としては、学習と開発環境構築の2つに分けることが出来ます。

・学習は、e-Learningとスクラム入門が中心となります。

・開発環境構築として、受講者のPC内にアプリケーション開発環境であるAndroid Studioの導入（クライアント開発環境構築）と、Webサーバのインストール（サーバ環境構築）により、各受講者がスタンドアローンにて開発出来る環境を構築します。

・講師側では、共通で使用するWebサーバの構築を前もって実施しておき、Android Studio上のサンプルソースからアクセスできる事を事前に確認しておきます。

2. アプリケーション開発実施

◯アプリケーション開発を円滑に進めるために、Redmineによるプロジェクト管理がとても重要となります。個々の受講者向けにRedmineのユーザIDを個別に作成し、配布する必要があります。またRedmineのユーザIDと受講者が常にアクセス出来るメールアドレスを紐付ける事により、Redmine上で関連する情報が更新された場合、登録されたメールアドレスに通知が送付されます。

◯アプリケーション開発には、必ず守らなければならない期間があります。スクラム開発において、この期間を厳守する事を厳格に求められます。 期間のはじめに開発する機能と担当者を決めて、開発を開始し、期間の終わりに開発した機能のレビューを実施して、レビューに合格した機能をユーザにリリースして、開発期間を終了します。

3. まとめと発表

◯すべての開発を終了した後に、これまでの作業内容についてまとめ、発表を実施します。発表は開発チーム毎に行います。

3. 開発手法(アジャイル)
3.1 どのような開発手法か

アジャイル(Agile) = 敏捷(びんしょう)、すばやいという意味

4つの基本理念

1. プロセスやツールよりも人と人同士の相互作用を重視する。
2. 包括的なドキュメントよりも動作するソフトウェアを重視する。
3. 契約上の交渉よりも顧客との協調を重視する。
4. 計画に従うことよりも改変に対応する事を重視する。

最も重要な事は顧客を満足させ、早く継続的に価値あるソフトウェアをリリースする。

数週間、数ヶ月単位で頻繁に実用的なソフトウェアをリリースする。タイムスケールは短いほど良い。

開発の終盤においても要求の変更を受け入れる。アジャイルは顧客の競争力を優位にする為の道具。

8

◯このカリキュラムでは、スクラム開発手法によるアプリケーション開発を実施しますが、スクラム開発手法は大きく分けて、アジャイル開発手法の1種として分類されています。このアジャイル開発手法と対になる開発手法として、古くからソフトウェア開発会社で使われてきているウォータフォール開発手法があります。ウォータフォール開発では、要件定義、外部設計、内部設計、実装・単体テスト、結合テスト、システムテストなどのフェーズに分けて上流工程から下流工程に順番に作業を実施して行きます。工程ごとに作業が分割されており、長い歴史の中でプロジェクトの管理方法などが確立されています。この手法の欠点は、下流工程において発生した変更点を上流工程に遡って反映する事が難しく、工数増大の要因になる事です。
他方、アジャイル開発は比較的少人数での開発に向いており、短めの開発サイクルや分割リリースなど、仕様変更などの変化に対応しやすい特徴があります。

◯アジャイル開発手法や、ウォータフォール開発手法は、どちらが良いと言うものでは無く、開発対象やメンバー、ユーザの状況により、最適な開発を実施するための選択肢と考えます。

3. 開発手法(アジャイル)
3.2 スクラム開発技法(scrum)

今回の開発において、スクラム開発技法をプロジェクト管理のフレームワークとして使用します。

スクラム開発技法(scrum)とは、アジャイル開発の手法の一つで、比較的簡単に学習できて、効果を出しやすい事が特徴。

スクラムとは、スポーツのラグビーの様に開発チームがゴールまで塊となって進む開発手法で、ウォーターフォール開発は、リレー競走に例えられる。

今回はタイムスケール＝4週間を1単位として進める。

(1単位(4週間) ＝ 1スプリントと呼びます。)

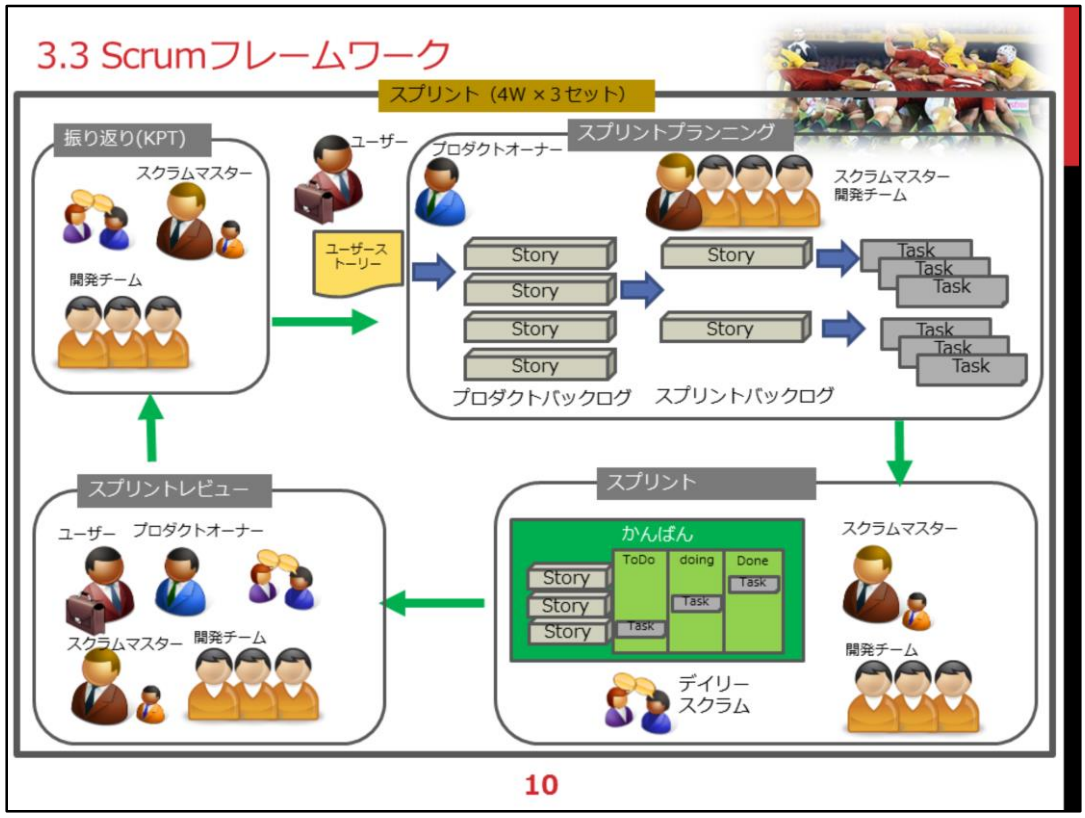
※ スクラム開発の詳細は、事前学習のスクラム入門(基礎編、実習編)を参照して下さい。

9

◯スクラム開発手法とは、ラグビーのスクラムを語源にとしている開発手法で、プロジェクト管理に特化した開発手法です。開発メンバーが自分の特技を活かして、コミュニケーションをしっかりと取ることで最大限の開発効率を発揮する事が出来ます。チームのメンバーは、4～10名位と比較的小規模な開発に向いています。

◯スプリントという開発の期間を定めて、その期間を厳守して開発を進めていきます。今回の開発では、1スプリント＝4週間として、全3スプリント実施します。

◯スクラム開発手法を学ぶために、スクラム実習(スクラムシミュレーション)を受講して下さい。資料は、2.10_Scrum_Simulation1_Introduction_jpn.pptxと、2.20_Scrum_Simulation2_Practice_jpn.pptxを使用します。またレゴブロックがあれば、楽しく効果的にこのシミュレーションを体験し、スクラム開発手法を学ぶことが出来ます。スクラムシミュレーションの実施時間は、約4～5時間です。



◯スクラムフレームワークには、スプリントプランニング、スプリント、スプリントレビュー、振り返り（レトロスペクティブ）が順番に実施されます。

・ユーザの要望をユーザーストーリーとして記述されることからはじまります。この作成されたユーザーストーリーに優先順位を付与したものをプロダクトバックログと呼びます。

・スプリントプランニングにおいて、今回のスプリントで開発する機能をプロダクトバックログから選択し、スプリントバックログとして抽出します。これらスプリントバックログを開発メンバーが作業単位であるタスクに分解します。

・スプリントにおいて、実際の開発は作成されたタスクを開発メンバーにアサインして、開発を実施します。アサインしたタスクが完了した後に次のタスクをアサインします。この期間において、毎日ディリースクラムと言われる作業報告を実施します。また作業の状況は“カンバン”（TaskBoard)上で管理します。

・スプリントの終盤に、スプリントレビューとして、それまで開発した機能を関係者全員でレビューします。このレビューで許可されたもののみが当スプリントでリリースされる機能になります。

・スプリントの最後に、当スプリント全体を振り返って次のスプリントがより良く実施出来る様に準備します。これをレトロスペクティブと呼びます。

3. 開発手法(アジャイル)

3.4 スクラムの道具

【3つのロール(役割)】

(1)プロダクトオーナー・・・製品の責任者

(2)スクラムマスター　・・・スクラムの番人、チームを守る

(3)開発チーム　　・・・プロジェクトにフルタイム参加する開発メンバー
プログラマ、アーキテクト、デザイナーの集合

❖当プロジェクトでは、スクラムマスターをアドバイザー(日本＆海外)が担当し、開発プロジェクトを支える。

【4つのミーティング】

(1)スプリント計画会議・・・スプリントのゴールとスプリントバックログを決定する

(2)ディリースクラム　　・・・毎日15分の報告会(3つの質問)

(3)スプリントレビュー　　・・・完成したアプリのレビューを実施

(4)ふりかえり　　・・・うまくいった事、いかなかった事などを報告

(Retrospective)　　(Keep:継続すること, Problem:やめること, Try:やること)

❖ 当プロジェクトでは、スクラムのプロジェクト管理全体をRedmineにて管理する。

11

◯3つのロール

・スクラム開発においては、この3つのロールのどこかに開発メンバーが含まれます。ほとんどの開発メンバーは、開発チームの役割を任されます。

◯4つのミーティング

・スクラム開発において必ず実施しなければならないミーティングがこの4つです。これらの実施結果は、必ずRedmine上に登録して記録を残すようにします。

・特に重要なミーティングが、ディリースクラムで毎日決まった時間に15分間程で実施ます。

・一番難しいミーティングは、スプリント計画会議(スプリントプランニング)になります。

3. 開発手法(アジャイル)

3.5 スクラムの道具

【3つの道具】

(1)プロダクトバックログ ・・・顧客要望をユーザストーリーとして一覧化した
Product Backlog　　ものでプロダクトオーナーが優先順位を付ける

(2)スプリントバックログ ・・・スプリントの対象となるプロダクトバックログ項目
Sprint Backlog　　をタスク化したもの

(3)バーンダウンチャート ・・・今どれ位すすんでいるか、どれ位で完了するか
Burndown Chart　　をグラフ化したもの

- 当プロジェクトでは、上記3つの道具をRedmine上で運用し管理する。
- スクラム開発手法については、スクラム入門(基礎編、実習編)にて学びます。

12

◯スクラム開発を滞り無く実施していくための道具を指します。
・プロダクトバックログ　ユーザー要望に優先順位をつけて管理しているもので、これをすべて完了すれば、プロジェクトの完了になる。
・スプリントバックログ　現在実行しているスプリントで開発対象とするプロダクトバックログの事で、無理なくスプリント内で終了できる範囲を見積り対象とします。
・バーンダウンチャート　今回の実習では利用しなくても良い。Redmineではバーンダウンチャートを自動生成してくれる機能があります。

4. 設計から開発の手順
4.1 作業のながれ

1. スプリント(4週間)開始前に、4つのイベント(スプリントプランニング、ディリースクラム、スプリントレビュー、スプリントレトロスペクティブ)の実施日程を開発メンバーで打合せて確定し、必ず守る。(海外側との確認・調整を忘れずに)
2. ユーザ要件を理解し、ユーザストーリーを作成する。
3. ユーザストーリーをプロダクトオーナーがRedmine上に優先順位をつけて登録する。(プロダクトバックログの登録)
4. 開発チームがプロダクトバックログから、今回のスプリントで開発する対象のストーリーを選択し優先順位をつけてリスト化する。この際に詳細な仕様をRedmine上で登録する。また作業量を見積る。(スプリントバックログの登録とスプリントプランニングの実施)
5. 開発チームがスプリントバックログを基に細分化してタスクリストをRedmine上登録する。(タスクの登録) チームのメンバーは、登録されたタスクを自分自身でアサインして開発する準備を行う。
6. スプリントが実施される。毎日ディリースクラムを実施して結果をRedmineに登録する。チームから挙がってくるサーバ側の変更要望(データベースやインターフェース)をチーム内の担当者が判断して実装する。発生した課題は、障害チケットとしてRedmineに登録し管理する。

13

◯実際に開発を行う際の手順を説明しています。まずはイベントを実施する日程を決めて、次に要望をユーザーストーリーに分解して優先順位をつけます。優先順位はプロダクトオーナーのみが変更できる作業です。この機能一覧をプロダクトバックログと呼びます。

・プロダクトバックログから各スプリントで開発対象とするバックログを抽出したものをスプリントバックログと呼びます。スプリントプランニングミーティングにてこの抽出作業を行ってスプリントでの開発範囲を決めます。

4. 設計から開発の手順
4.2 作業のながれ

7. スプリント終了直前に、チームはスプリントレビューを実施してレビュー結果をRedmineに登録し、プロダクトオーナーが確認する。バグは、BugチケットとしてRedmineに登録し管理する。
8. チームは、スプリントレトロスペクティブ（ふりかえり）を実施して、結果をRedmineに登録する。
9. リリースされたソフトウェアをプロダクトオーナーとユーザ（日本側アドバイザー）で確認し、Bugが発生した場合、Redmine上のチケットを登録して管理する。
10. 次のスプリントの準備を開始する。プロダクトオーナは、スプリント完了時の残作業を確認し、プロダクトバックログの優先順位を更新する。
11. その後、次回スプリントの開発対象を抽出して見積り、タスク化する作業からスプリントを開始する。（スプリントプランニングの実施）

14

◯開発を終了したスプリントバックログを元にスプリントレビューを実施します。レビューには、開発関係者及びユーザが参加します。レビューで許可を得た機能は、当スプリントにおいて、リリース対象とされて、抽出されユーザ側に納品します。

4. 設計から開発の手順
4.3 会議(イベント)の見える化

あらかじめ会議を予定しておき調整の無駄を避ける。(海外側にも確認要)
スプリント途中で会議スケジュールは絶対に変更しない。

	Mon	Tue	Wed	Thu	Fri		M	T	W	T	F		M	T	W	T	F		Mon	Tue	Wed	Thu	Fri
AM		Sprint Planning																AM					
PM 1																		PM1	Sprint Review				
PM 2																		PM2	Retrospective				
Ex																		Ex			Release		

4 Weeks Sprint

スプリント計画 (Sprint Planning) ・・・ 開始するスプリントで実装する内容を計画
4時間/1スプリントを基準とする
スプリントレビュー (Sprint Review) ・・・ 実装した機能のレビューとリリースの判断
2時間/1スプリントを基準とする
ふりかえり (Retrospective) ・・・ スプリントをふりかえって、改善策を提案＆実施する
2時間/1スプリントを基準とする
リリース (Release) ・・・ 開発＆レビュー済アプリケーションをリリースする

15

◯当カリキュラムでは、1スプリント＝4週間と決めているので、4週間でスプリント計画（スプリントプランニング）～リリースまでをどのようなスケジュールで実施するかをスプリントの最初に決めております。

5. 顧客要望
5.1 顧客要望からユーザストーリーへ

日本側チームは、プロダクトオーナーが中心となり、顧客要望一覧からユーザストーリーを作成して、プロダクトバックログを作成する必要がある。

ユーザストーリーとは

- 要件（機能）をスケジュール可能な単位に分解したもの
- ○○(誰が)として、◇◇(何を)をしたい、なぜなら△△だからと表現する
- 他の要件に依存しない状態に分解する
- ユーザがどう使うかという目線で表現する（ユーザの言葉で書く）
- 実際に動作する機能単位で分割する
- 受入テストを記述できるレベルに分割する（受入れ基準がある）
- プロダクトオーナーが優先順位をつける

※ 実装済み機能のプロダクトバックログは、登録済みなのでこれを参考に作成する。

16

○3.20_ユーザ要件.xlsxからユーザストーリーを抽出します。サンプルソースと比較してユーザストーリー内で既に実装済みのものを判別して、プロダクトバックログから除外しておきます。
・開発メンバー全員でユーザストーリーの開発優先順位を決める話し合いを実施します。最終的な決定権はプロダクトオーナーに有ります。

5. 顧客要望
5.2 ユーザ要望の見方

当プロジェクトのユーザ要望には2種類の状態がある。

ユーザ要望の全リスト

日本語版＝2.20_ユーザ要件.xlsx

英語版　＝2.21_User_Requirements.xlsx

①ユーザ要望に記載されており、すでにサンプルアプリケーションとして実装されている機能

②ユーザ要望に記載されており、まだ実装されていない機能

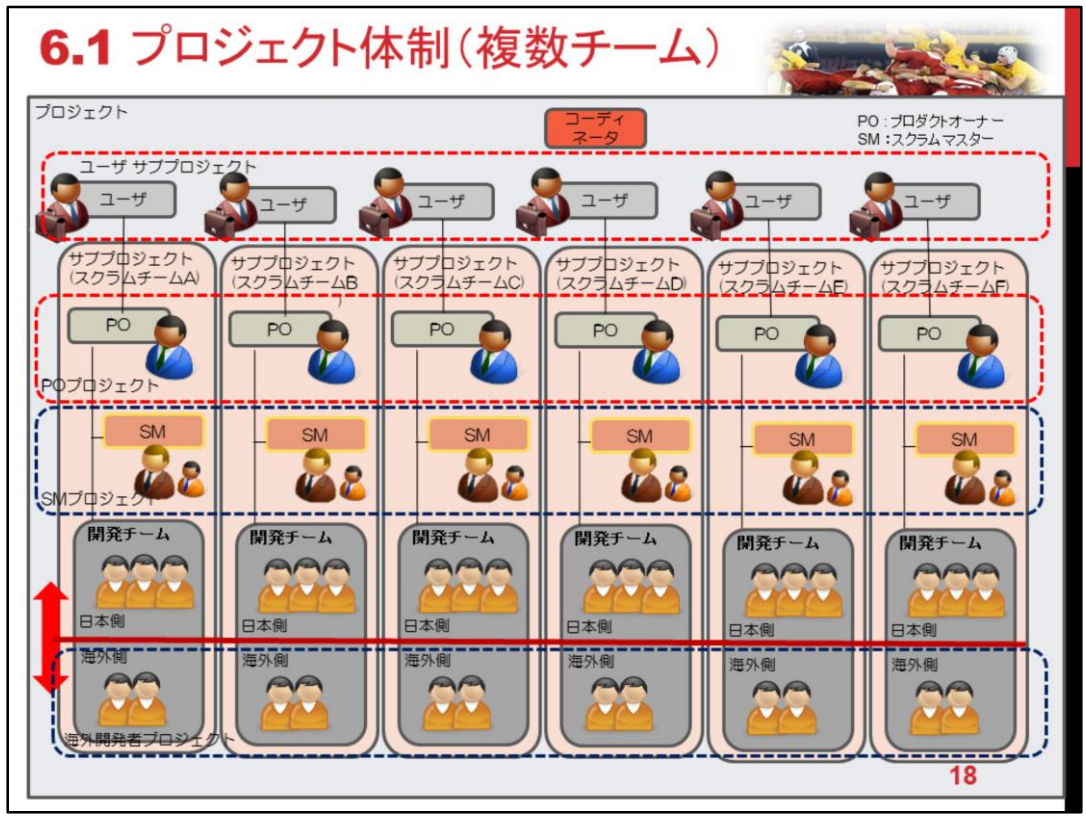
ユーザ要件ドキュメントを読んで、上記①と②を確認し、これから実装する機能のユーザストーリーリストを作成し、プロダクトバックログとして登録します。
また、サンプルで実装済みの機能についても拡張する事が出来る。

17

◯ユーザストーリーの作成には時間がかかりますが、ユーザ要件上の要件をもれなく満たしているかをよく確認して作業を実施して下さい。また既に実装済みのユーザストーリーについては、プロダクトバックログの対象から除外します。

◯ユーザストーリーの抽出が完了した後、優先順位を付与して、プロダクトバックログを完成させます。

◯上記の作業は、Redmine上で登録します。実際の操作については、3.10_Redmine使用方法(BackLogs).pptxを参照して下さい。



◯スライドは、6チームにて授業を実施した場合の例です。Redmine上には、開発チーム、プロダクトオーナー、スクラムマスターなどといった、情報交換できるプロジェクトを設けており、役割毎にチームを横断的にRedmineを通して打合せをすることが出来る様に構築し、活用します。

6.2 担当と役割

チーム / 役割		スクラムチームA	スクラムチームB	スクラムチームC	スクラムチームD	スクラムチームE	スクラムチームE
ユーザ		日本の講師	日本の講師	日本の講師	日本の講師	日本の講師	日本の講師
プロダクトオーナー		日本の学生	日本の学生	日本の学生	日本の学生	日本の学生	日本の学生
スクラムマスター	メイン	日本の講師	日本の講師	日本の講師	日本の講師	日本の講師	日本の講師
	サブ	海外の講師	海外の講師	海外の講師	海外の講師	海外の講師	海外の講師
開発メンバ		日本の学生	日本の学生	日本の学生	日本の学生	日本の学生	日本の学生
		海外の学生	海外の学生	海外の学生	海外の学生	海外の学生	海外の学生

※SM(サブ)：海外の講師は海外の学生のフォロー

◯この時点で、受講者を4～8名単位のチームに分けてください。チームに分けた後、各チームでプロダクトオーナー、スクラムマスター、開発メンバーの役割を決定し、このスライドに名前を記載して下さい。

6.3 プロジェクトにおける役割

役割		日本	海外	概要
ユーザー		○		開発するアプリケーションのユーザーストーリー（要望事項）を提示する人物。
プロダクトオーナー		○		開発するアプリケーションの仕様を決定する人物で、機能仕様上で問題や疑問点が発生した場合に、機能仕様を最終的に判断して決定する。
スクラムマスター	メイン	○		スプリントバックログが更新されているかチェックする人物で、チームの効率が最大になるように配慮し、問題が発生した場合の相談役としての役割も持つ。
	サブ		○	海外側のチームの効率が最大になるように配慮し、問題が発生した場合の相談役としての役割。
開発チーム		○	○	スプリントバックログの更新を行う人物で、実装とテストの能力を持つ。 開発チームには、下記の技術者を含みます。(兼務可) ・プログラム開発者　…機能を実装するプログラマ ・テスター　…テスト仕様書を作成し、製品をテストする ・システムアーキテクト …利用している技術とサーバ環境を保守する ・データベース管理者　…データベースの変更・運用管理者

20

◯上記の内、ユーザは講師が兼任します。受講者は、ユーザ以外のどれかの役割に必ず属します。チーム内の構成人数が少ない場合、スクラムマスターを設けない事も可能です。プロダクトオーナーは必ず設ける必要があります。
◯開発チーム内には、プログラマー、テスター、システムアーキテクト、データベース管理者など開発に必要な役割を割り当てます。ただし、肩書はすべて開発チームであり、誰でもどの役割でも兼務する事が出来ます。また開発中に有識者から方法を学んで役割を実施することも出来ます。

6.4 For the exercise course 2015
Team X members

Product Owner (JAPAN)
Mr. XXXX (3rd grade)
e-mail:

Skype ID
(Japan) :
(Oversea) :

Member (JAPAN)
Mr. XXXX (3rd grade)
e-mail:

Member (Oversea)
Mr. XXXX (Xrd grade)
e-mail:

Member (JAPAN)
Mr. XXXX (3rd grade)
e-mail:

Member (Oversea)
Mr. XXXX (Xrd grade)
e-mail:

Member (JAPAN)
Mr. XXXX (3rd grade)
e-mail:

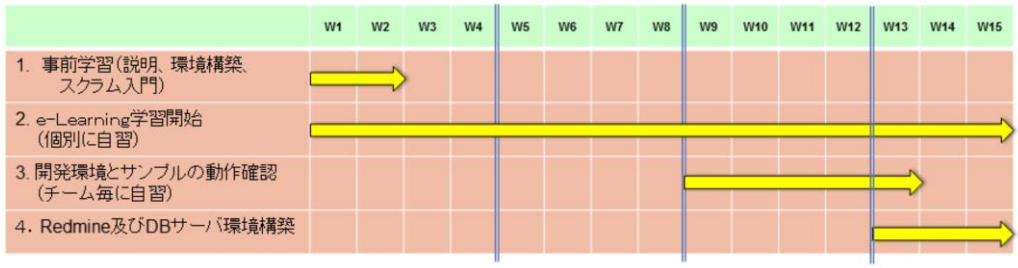
Scrum Master (JAPAN)
Mr. XXXX (Adviser)
e-mail:

Sub Scrum Master (Oversea)
Mr. XXXX (Adviser)
e-mail:

21

◯外国のエンジニアを協働開発する為に、各役割と氏名、SkypeIDなどを一覧にまとめた表を作成して、情報共有します。これらの情報共有資料は作成後、Redmineにアップロードして共有すると効率が良いです。

7. スケジュール
7.1 事前学習



事前学習のスケジュールの一例として上記を示している。
これは、スクラム開発の前に1コマ(90分)x15週の準備時間を取れる場合のサンプルスケジュールとなっている。
また、最小2週間の事前学習にて本編を開始することも出来る。

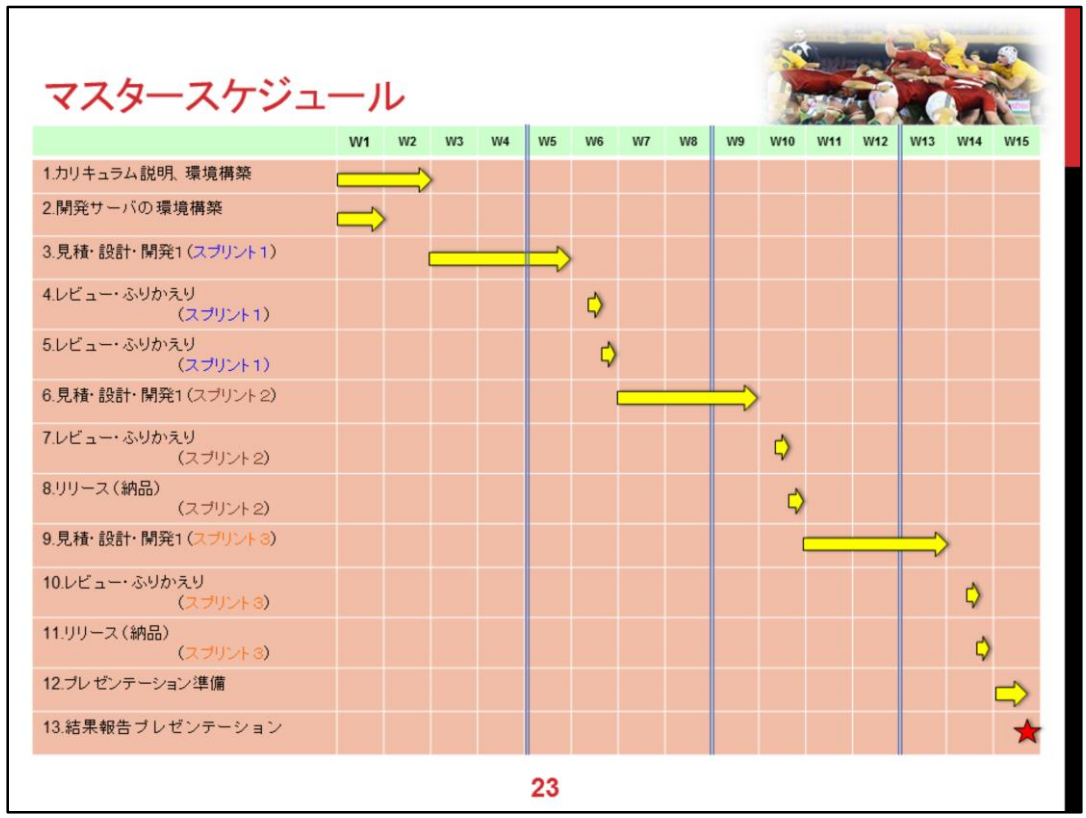
22

◯事前学習として、実施するもののスケジュール案です。本カリキュラムを実施する前にどれ位事前学習の時間を取ることが出来るかに合わせて調整して下さい。

・e-Learningは、本カリキュラム開始と同時に併行して実施するスケジュールにすることも出来ます。

・スクラム入門は、必ず本カリキュラム開始前までに実施する必要があります。

・講師は本カリキュラム開始前までに、WebサーバとRedmineサーバの構築と動作確認を完了させて下さい。



◯本カリキュラムのマススケジュールです。実際の日程等に合わせて調整して下さい。

8. 開発環境構築及び使い方
8.1 クライアント環境構築（インストール）

以下のドキュメントを参照

3.10_アンドロイド開発環境構築手順書.docx

または

3.20_アンドロイド開発環境構築手順書_簡易インストール版.docx

- 資料及びインストールするソフトウェア等一式は、DVDにて配布する。
  （今回使用する資料は、Redmineサーバからも取得する事ができる）
- 実行環境によっては、AVD側のProxy設定が必要となる。

24

◯全ての開発メンバーが個々に開発用PC内にインストールする必要があります。

◯簡易インストール版とは、開発用PCにインストールしたディレクトリをそのままコピーして環境を構築し、Android Studioの設定情報のみを調整する導入方法で、インストール時間を非常に短くする事が出来るが、コピー元とPC環境が大きく異なる場合には、正しく動作しない危険性があります。この方法で導入する場合、コピー元は約10GBのサイズになります。

9. 当プロジェクトのRedmine
9.1 Redmineの使い方

Redmineとは、Webベースのプロジェクト管理ツールで、世界中で利用されている。詳細については、以下のドキュメントを参照

2.10_Redmine使用方法(BackLogs).pptx

今回はRedmineサーバにスクラム開発管理用のBacklogsプラグインを導入した状態で利用します。

※ Redmine＋Backlogsサーバ構築方法については、別途環境構築用資料があります。

25

◯個々の受講者が、Redmineにログインして一連の使い方を学習します。これを受講する前までに受講者一人一人のユーザIDとパスワード、権限設定を完了しておく必要があります。

◯Excel等で、ユーザIDと受講者のマッチングリストを作成しておくことで、ログイン時のトラブルを抑えることが出来ます。

10. サーバ環境
10.1 テスト用サーバ環境構築(インストール)

個別でテストを実施する為にテスト用サーバ環境が必要な場合、下記のドキュメントを参考にして、Windows OS上にテスト用環境を構築する事ができる。

3.30_テスト用サーバ環境構築手順書(WindowsOS版).docx

インストールされる環境は、Apache、MySQL及びPHPと付属の管理ツール

構築されたXAMPP環境内に、今回使用するPHPスクリプトとDatabaseを設置する。

環境構築後、開発アプリケーション ソース内のアクセス先サーバURLを変更する事により、グローバル環境でのテストを実施する事ができる。

strings.xmlファイル内の各URL(http://10.0.2.2/・・・.php) を編集する。

(例 10.0.2.2 ➔ mail.tym-bit.comなど)

26

◯テスト用サーバとは、開発用クライアントPC内にWebサーバ＋データベースサーバを構築し、スタンドアローンにて開発～テストを行える為の環境です。手順に従って構築して下さい。
・テスト用サーバは、XAMPPという実行環境をベースに構築する方法を記載してあります。XAMPPの使い方等については、インターネットにて情報を取得して下さい。

◯はじめから共通のサーバを使用する場合(講師が予め構築して置きます)は、当作業は必要ありません。

10. サーバ環境
10.2 サーバ環境(データベース及びスクリプト)

サーバ環境内に以下のデータベース及び、PHPスクリプトがインストールされている。但し、これらはスプリント0(初期設定)状態なので、アプリケーション開発に合わせて各チームで拡張して利用する必要がある。

データベース仕様書(MySQL用)

2.50_データベース仕様書.docx を参照。

Database名: mapdb, User名: mapdb, Password: ****

インターフェース仕様書

2.60_サーバ端末間インタフェース仕様書.docx を参照。

27

◯データベース仕様書とは、配布されたサンプルソースが使用しているデータベースのテーブル構造のドキュメントを指しています。

◯機能を拡張し、データベースのテーブルを追加・変更する場合は、このドキュメントにも変更内容を反映するように伝えて下さい。

10. サーバ環境
10.3 サーバ環境(グローバルアクセス)

インターネットからアクセスできるサーバ環境(本番)は、以下の構成となっている。
下記のサーバ環境を利用するかどうかは、チームで決定する事

- OS: CentOS 6.5 (Redhat linux clone)
- Global IP: グローバルIPアドレス
- Local IP: ローカルIPアドレス
- Using port: HTTP(80), HTTPS(443), SSH(22), VNC(5901-5902)
  - ※ SSH及びVNCはアクセス元を限定している

XAMPP for Linux ver5.6.12

- Apache 2.4.16, PHP 5.6.12, phpMyAdmin 4.4.14
- MySQL 5.6.26
  - 管理者ID: root, Password: ****
  - Database名 : mapdb_a~f, User名: mapdb, Password: mapdb

MySQLサーバにアクセスする場合、VNCviewerを使ってログインし、Webブラウザから "http://localhost/phpmyadmin/"にアクセスする。
これらの環境を使用する際は、問い合せて下さい。

28

◯講師側で、本カリキュラムの共通Webサーバを構築した場合のサーバに関する情報(例)です。実際に構築した内容に合わせて変更して下さい。

11. e-Learningの使い方

今回の実証講座では、株式会社デジタル・ナレッジ様から提供されたe-Learningシステムも併用します。このe-Learningシステムは、当講座に対する理解を深めるために使われます。

特に開発における海外の学生とコミュニケーションを取るためのノウハウが詰まっているので、何度も利用して下さい。

また講座実施中に発生した疑問や問題点に対するサポートもこのe-Learningシステムを使います。

利用者は個々にID＆パスワードを受取り、アクセスして下さい。

アクセス先URL: https://deli3.study.jp

29

◯本カリキュラムを円滑に実施する為に、e-Learningによる事前学習はとても重要です。受講することにより、スクラム開発手法による開発時の英語表現やプレゼンテーション資料の作成方法、スクラム概要をいつでも簡単に学習することが出来ます。また、繰返し学習することで習得効果が高くなります。

おわりに

当講座で、アプリケーション開発を行いますが、**その完成度に固執しない**で下さい。

目的は、**スクラム開発技法を理解**し、実習を通して確実に実行し、どのような長所や短所があるかを理解することです。

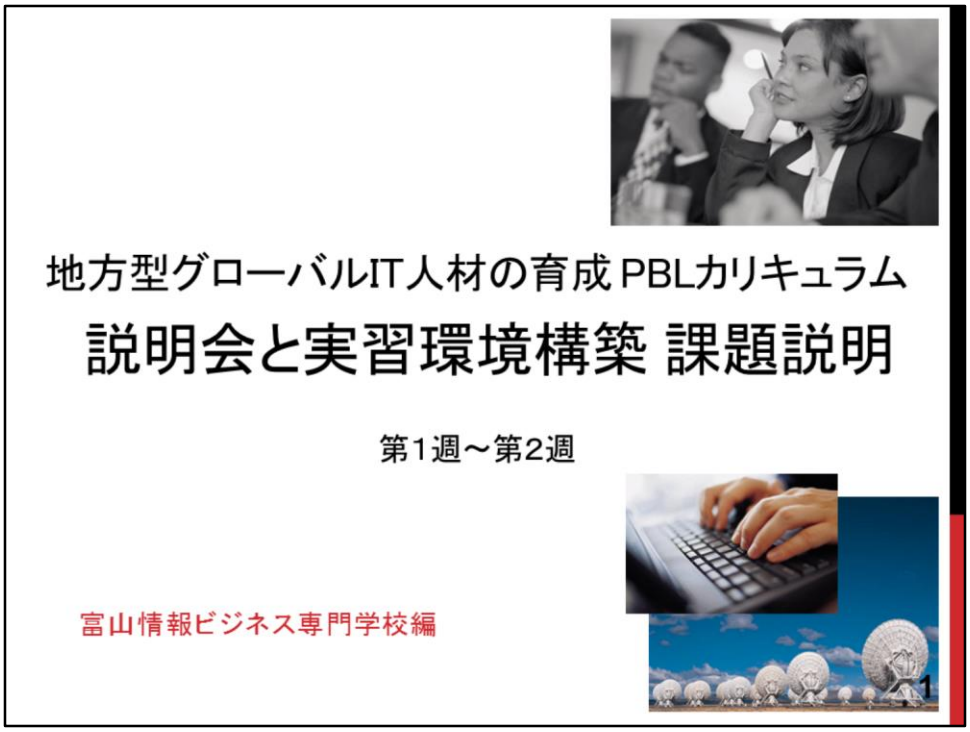
また開発メンバーが同じ目的に向かって、開発を**全員で協力し、楽しみながら進める事**が重要で、この目的の為にプロダクトオーナーとスクラムマスターは、特に尽力して下さい。

海外メンバーを含めてコミュニケーションを密接に取って納得しながらプロジェクトを進めて下さい。ゲームやアニメを含む文化や習慣の違いなど**プロジェクト以外についてもどんどん意見交換**して下さい。

当講座では、複数チームが同じ課題を実施しますが、決して**競争ではありません**。

30

◯複数のチームに分かれて本カリキュラムを実施する場合の注意事項です。目的はスクラム開発手法を習得し、国内エンジニア及び、海外エンジニアと円滑にコミュニケーションを取りながら開発を進める方法を学ぶことなので、他のチームと競い合う必要が無い事を常に伝えて下さい。他のチームメンバーは、競争相手ではなく、スキルや情報を共有してお互いによりよいアプリケーションを開発するための仲間と考えて下さい。



1. 目的

・当ドキュメントにおいて、開発に必要な開発環境を受講者のPCにインストールして、正しく動作することを確認します。

・当カリキュラムの受講者の評価基準を説明し、授業としてどのような観点で評価されるかを理解させます。

2. 教材

・必要に応じて下記の教材を使用します。

3.10_アンドロイド開発環境構築手順書.docx
3.20_アンドロイド開発環境構築手順書_簡易インストール版.docx
3.30_テスト用サーバ環境構築手順書(WindowsOS版).docx

3.11_Instruction of the quick Building development environment.docx
3.21_Instruction of the quick Building development environment.docx
3.31_Instruction of Test server environment on WindowsOS.docx

上記3.10、3.20、3.30は日本語で記述された教材で、3.11、3.21、3.31は英語で記述された教材となっており、受講者の状況に応じて利用して下さい。
また3.20 簡易インストール版(3.21 英語版も同様)とは、インストール先のPCやネットワーク環境があまり整っていない場合に使用するインストール手順書です。通常のインストール手順を実行するとインストール中に2～3GByteのファイルをインターネットからダウンロードされます。海外などネットワーク

の環境があまり整っていない場合は、インストールに数時間かかる場合があります。その為、講師側で予め3.10 アンドロイド開発環境構築手順書に従って通常のインストール作業を実施し、完成したインストール環境をそのまま受講者PCにコピーして、環境設定ファイルのみ更新する事で大幅にインストール時間を短縮することが出来ます。インストール済みの環境をコピーする場合、ファイルサイズ＝約10GByteのコピーが必要となります。USBメモリなどコピー方法をあらかじめ検討しておいて下さい。

進め方

＜＜　課題　＞＞
・教師より当授業の目的と進め方の説明を受ける。
・当授業に関する必要なドキュメントの配布を受け、その内容を熟読し理解する。
・手順書に従って実習環境を構築する。
・実習環境でインストールされた各種ツールの使い方を学ぶ。
・ユーザー要件を熟読し、システムの全体像をイメージする。

＜＜　盛り込む内容　＞＞
(1)対象となるシステム開発プロジェクトを理解して、グループと役割を決める。
・要員構成
　○日本側(元請)：プロダクトオーナー、スクラムマスター、チームメンバー
　○オフショア側：チームメンバースクラムマスター、プログラマー
　○教師　　：顧客(システム開発の依頼者)、状況によってはスクラムマスターを兼務
　　※チームメンバーはシステムアーキテクト、DBアーキテクト、プログラマーなどを兼務

(2)実習環境の構築と使い方の学習
・検証環境　　構築手順書に従ってプログラム開発環境を構築する。
・管理ツール環境　構築手順書に従ってプロジェクト管理ツール環境を構築する。
・Webサーバ環境　構築手順書に従ってWebサーバ・データベースサーバ環境を構築する。
・プロジェクト管理ツールを使ってどのような管理ができるか、またどうやってオフショア会社側と意思を疎通させるかを考えながら使い方を学習する。

◯ここでは下記の作業を行い、開発環境の整備とRedmineの基本的な使い方を学びます。
　・受講者のPCに、開発環境一式をインストールします。
　・受講者のPCに、サーバ環境一式をインストールします。上記と合わせてサンプルソースが正しく実行されることを確認します。
　・当カリキュラム全般で使用するRedmineにログインするためのユーザIDとパスワードを受講者に配布し、Redmineの使い方を学習します。
　・クライアント開発環境、Webサーバ環境とRedmineがどのように連携して、アプリケーション開発を進めていくかを学習します。

◯これから一緒にアプリケーション開発を行うチーム分けをします。チーム分けの際は、受講者のスキルが特定のチームに偏らないように考慮して下さい。特にJava言語による開発スキルとデータベース操作のスキルを持った受講者が、各チームに均等に配置されるようにすると円滑な開発を行うことが出来ます。

◯チーム分けが完了した後に、チーム内の役割分担を行います。
・プロダクトオーナー ・・・ 1名選出して下さい。アプリケーション開発の知識はあまり必要ありません。ユーザの代理として開発する機能を管理し、開発の優先順位を決定する権利を持っています。
・開発メンバー(チームメンバー) ・・・ 実際のアプリケーション開発を実際に行うメンバーです。日本メンバー、海外メンバーのほとんどがこの役割に属しま

す。
・スクラムマスター ・・・ 基本的にアプリケーション開発に参加しないで、スクラム開発を外側からささえる役割です。最もスクラム開発技法を深く知っている必要があります。チーム内の人数によってはこの役割を使用しない事も可能です。また、講師が兼任する方法もあります。
・ユーザ（顧客） ・・・ アプリケーション開発を依頼している顧客です。教師がこの役割を担当します。ユーザはプロダクトオーナーと開発する機能について話し合って、機能の実現方法や開発の優先順位をプロダクトオーナーに伝えて、情報を共有します。

進め方

<< 盛り込む内容 >>

環境構築に使用するテキスト
3.10_アンドロイド開発環境構築手順書
3.20_アンドロイド開発環境構築_簡易インストール版
3.30_テスト用サーバ環境構築手順書

※ 英語版の手順書が必要な場合、下記の手順書を使用する
3.11_Instruction of Building development environment
3.21_Instruction of the quick Building development environment
3.31_Instruction of Test server environment on WindowsOS

(3)評価と指標について理解する。
この実習では、教師による理解度の評価が実施される。(スプリント毎、計３回)

評価の為の情報は、学生のスクラム開発への参加度合いとオフショア側とのコミュニケーション及び、担当している作業内容の理解度を基本とします。これらの情報は、スプリント内の学生の言動や、Redmine上の受答えと報告実績及び、教師による質疑応答によって収集します。
授業で実施されるプロジェクトは、スプリント単位に開発対象の選出(バックログ)と機能設計、開発、振返り、納品及び、次スプリント準備のフェーズに分かれており、それぞれのフェーズにおける役割の理解と実施状況を判断します。また評価者は教師のみとし、全3スプリントの合計を最終評価とします。

◯ここでは、受講者のPCにアプリケーション開発環境のインストールを実施します。インストール方法としては、通常のインストールと簡易インストールのどちらかを講師が選定して利用します。どちらのインストール方法を採用するかの判断基準としては、アプリケーション開発を行う環境からインターネットにアクセスしやすいかどうかが重要なポイントになります。また外国人向けに英語版の教材を適宜利用して下さい。

進め方 (理解度の評価)

○5つの評価項目:

1.目的の理解度 …指定された役割で要求されている事は何でどのような結果を求められているかについて正確に把握できているかどうか。

2.作業の実行度 …指示された役割の目的達成に向けて正確に作業を実行できたかどうか。

3.問題解決力 …質問や発生した問題に対して、プロジェクトがより良い方向に向かう事を考慮して判断し、的確に行動できたかどうか。

4.コミュニケーション力…元請側で学生同士及び、オフショア側とのコミュニケーションが円滑に正しく行われているかどうか。

5.ツール理解度 …Redmineの使い方を正確に理解し、効率よく使用できるかどうか。

○6つの指標:

5(最も高い) … 評価項目を理解して実施し、目的通りの結果を得る事ができる。
4 … 評価項目を理解して実施し、目的に対して80%の結果を得る事ができる。
3 … 評価項目を理解して実施し、単独で50%の結果を得る事ができる。
2 … 評価項目を理解しているが、ほとんどが他者からの指示で実施できる。
1 … 評価項目をほとんど理解しないで、プロジェクトに参加しているのみ。
0(最も低い) … 全く評価項目を理解できず、進展も見られず成果物を提出できない。

◯ここでは、受講者の理解度の評価方法を説明します。5つの評価項目それぞれを5段階で評価します。評価項目は、アプリケーションの開発そのものでは無く、コミュニケーションの取り方や率先して作業に参加しているか、Redmineをうまく利用できているかに評価の重点が置かれている事を受講者に説明します。

◯評価は、各スプリント毎に実施して評価結果を次のスプリントが開始された時点で受講生に通知します。通知した結果を元に次のスプリントではどのような課題に対してどのように取組むかを明確にしてスプリントを実施できるように個々の受講生に指導します。

進め方 (理解度の評価)

○得点について

学生が得る得点は、上記5つの評価項目をそれぞれ6つの指標(0～5)に照らし合わせて評価するが、評価対象のスプリント(1～3)によって満点が以下の様に異なる。最終的な得点は3つのスプリントの合計によって求められる。これはスプリントを繰り返すことで理解度が上がり、より高度な作業を実施できる様になる事を想定してスプリント毎の得点重み付けを変化させている。

スプリント1 ・・・25点満点 (1指標ポイント＝1点)
スプリント2 ・・・25点満点 (1指標ポイント＝1点)
スプリント3 ・・・50点満点 (1指標ポイント＝2点)

(例 スプリント1の評価合計＝15ポイント ×1点 ＝15点
スプリント2の評価合計＝20ポイント ×1点 ＝20点
スプリント3の評価合計＝22ポイント ×2点 ＝44点
合計 ＝79点

○当カリキュラム全体を通した評価を実施する為に、各スプリント毎の評価を元にその重み付けを付与して最終的な評価を実施します。スプリントが進んでいくに従って、スクラム開発手法を理解し効率的に実施できることから、全体評価の重み付けは、スプリント1及びスプリント2はスプリント評価x1倍とし、スプリント3はスプリント評価x2倍として実施します。重み付けしたスプリント1・2・3の評価合計を最終評価とします。全体評価は、100点満点とし60点以上で当カリキュラムを合格とします。

○上記の全体評価は、1つの基本的な基準です。評価方法等については当カリキュラムを実施される講師の方が自由にアレンジして下さい。
・アプリケーション開発そのものを評価対象に加える事も出来ます。
・スプリント毎の評価を受講者に自己評価させて、講師側の評価と自己評価の差異を確認する方法も有効です。

◯この章では、事前講座としてスクラム開発手法に関する学習を説明します。はじめに座学でスクラムに関して概要を学び、次に簡単なプロジェクトを通してスクラムを実習形式で学びます。

・ここでのスクラム学習をスクラムシミュレーションと呼びます。また別名としてレゴスクラムとも呼ばれるように、当講座を実施する際にレゴブロックを活用すると更に効果的に楽しくスクラム開発手法を学習出来ます。レゴブロックを手配できない場合、厚紙と色鉛筆を使っても実習できます。

◯スクラムシミュレーション実施には、下記の2種類の教材を使用します。

Scrum_Simulation1_Introduction_jpn.pptx ・・・ 座学用スクラム学習 基礎編

Scrum_Simulation2_Practice_jpn.pptx ・・・ スクラムシミュレーション 実習編

また講師向けの説明が付与されている教材もあります。

2.10_Scrum_Simulation1_Introduction_jpn_教師向け.pptx

2.20_Scrum_Simulation2_Practice_jpn_教師向け.pptx

海外のエンジニア向け実施用の英語版もあります。

2.11_Scrum_Simulation1_Introduction_eng.pptx

2.21_ Scrum_Simulation2_Practice_eng.pptx

進め方

<< 課題 >>

・アジャイル開発の１種であるスクラム開発手法を理解する為に、本来プロジェクトが実行される前の準備段階でスクラム模擬開発の体験学習を実施する。これらを元受会社及びオフショア会社側で行うことにより、スクラム開発手法及び、当講座の進め方を理解する。また元請会社側（日本側）とオフショア会社側での作業内容について理解する。

<< 盛り込む内容 >>

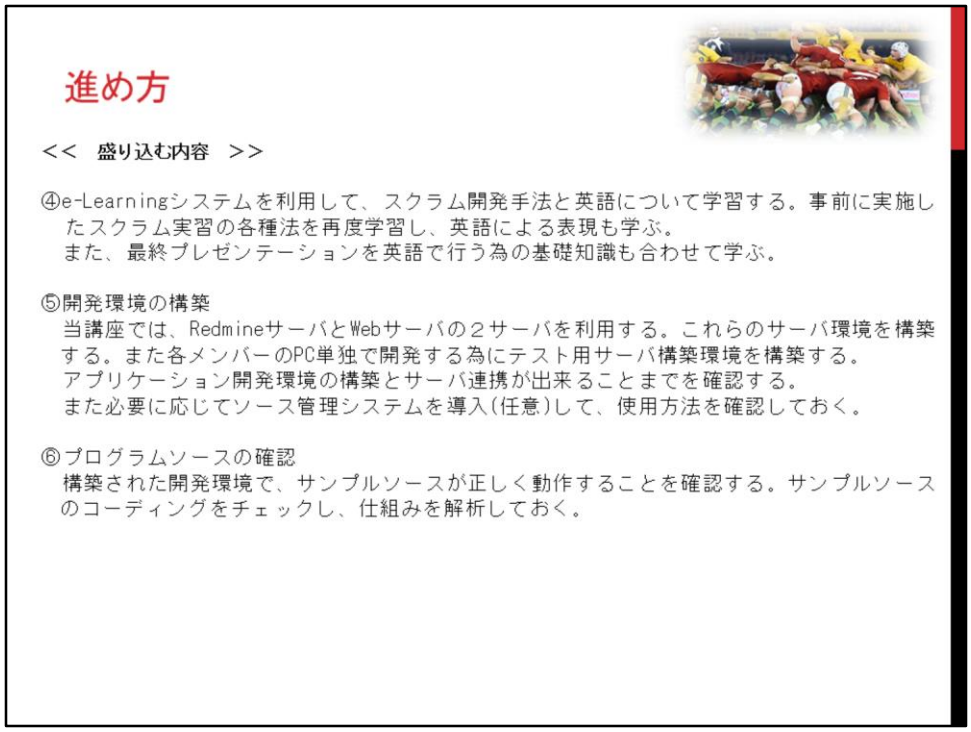
①役割の決定
教師が顧客及びプロダクトオーナーとなり、街作りを題材にスクラム開発手法による作業を示指する。この時の役割としてスクラムマスターと開発メンバーを決定する。

②座学としてスクラム開発の基礎を学ぶ。
スクラムの用語や仕組みを最小限の知識として学ぶ。

③実習を通してスクラム開発を学ぶ。
レゴブロックや厚紙を使って、模造紙の上に街を開発する。この際、スクラム開発に必要なバックログやタスクの意味やタスクボードの使い方を学び、スプリントプランニング、ディリースクラム、スプリントレビュー及び、レトロスペクティブを体験する。

◯スクラムシミュレーション実習では、講師がプロダクトオーナーとなり、プロダクトオーナーが住む街作りがプロジェクトのテーマとなります。街作りを模造紙、色鉛筆、厚紙、レゴブロックなどを使用して、想像力を働かせて行って下さい。場合によっては手持ちのスマートフォンや付箋紙などを追加で使用しても面白いものが出来ると思います。スクラムシミュレーションは楽しみながら、スクラム開発手法のエッセンスを習得できる教材で、この教材のみ単独で取り組みことも出来ます。

◯このシミュレーションもチームで実施するのですが、1チーム6名前後が理想的なチーム人数です。これは本番のアプリケーション開発でも同様です。1チームだけでは成果物を見せ合う張り合いが無いので、多少少な目の人数になっても、最低2チームで実施されることを推奨します。（もちろん1チームでの実施でも問題ありません）このチーム内には、開発メンバーとスクラムマスターの役割を設けますが、チームの人数によってはスクラムマスターの役割を省略することも出来ます。



◯事前授業は、スクラムシミュレーションとe-Learningの両方から構成されており、どちらからはじめても、同時に着手しても同様な効果を得ることが出来ます。

◯e-Learningシステムは、受講者個人毎にその学習のスピードと、問題点が異なるため、個々の受講者毎にユーザIDを作成して、受講されることを推奨します。特にスクラム開発で使用される英文例題等はアプリケーション開発中も良く参照されますので、当カリキュラム受講中はいつでもe-Learningシステムにアクセス出来る様に環境を整えておいて下さい。

◯開発環境の構築とは、受講者が使用するアプリケーション開発環境一式の構築を指します。クライアント環境構築には、3.10_アンドロイド開発環境構築手順書.docx（英語版は、3.11_Instruction of the quick Building development environment.docx）を参照して下さい。インストール時のインターネットアクセスを含むネットワーク環境が不安定な場合は、3.20_アンドロイド開発環境構築手順書_簡易インストール版.docx（英語版は、3.21_Instruction of the quick Building development environment.docx）を使用して環境構築を行うことも出来ます。また、ローカルPC内でWebサーバの機能も導入し、スタンドアローンでの開発環境一式を構築する場合は、3.30_テスト用サーバ環境構築手順書(WindowsOS版).docx（英語版は、3.31_Instruction of Test server environment on WindowsOS.docx）を使って構築して下さい。これらは、いづれもWindows7及び、Windows8用に記載されており、Windows10へのインストールを実施する際は、ファイアーウォール等セキュリティ内容を変更する必要があります。特

にテスト用サーバ環境構築の際は、80番ポートが既に他のアプリケーションで使用されている為、何らかの対策が必要になります。
次にRedmineへのアクセス方法と使い方を学習します。資料は、2.10_Redmine使用方法(BackLogs).pptx（英語版は、2.11_How to use Redmine(+Backlogs).pptx)を使用して下さい。

◯もし開発したソースコードの管理を簡略化する場合、GitやSVCなどのソース管理システムの導入と利用をお薦めします。Android Studioはこれらソース管理システムとの連携を考慮されています。

理解度の確認

下記の質問に対して答えなさい

問１．スクラム開発とはどのような開発手法か？またどのようなツールを使用するか？

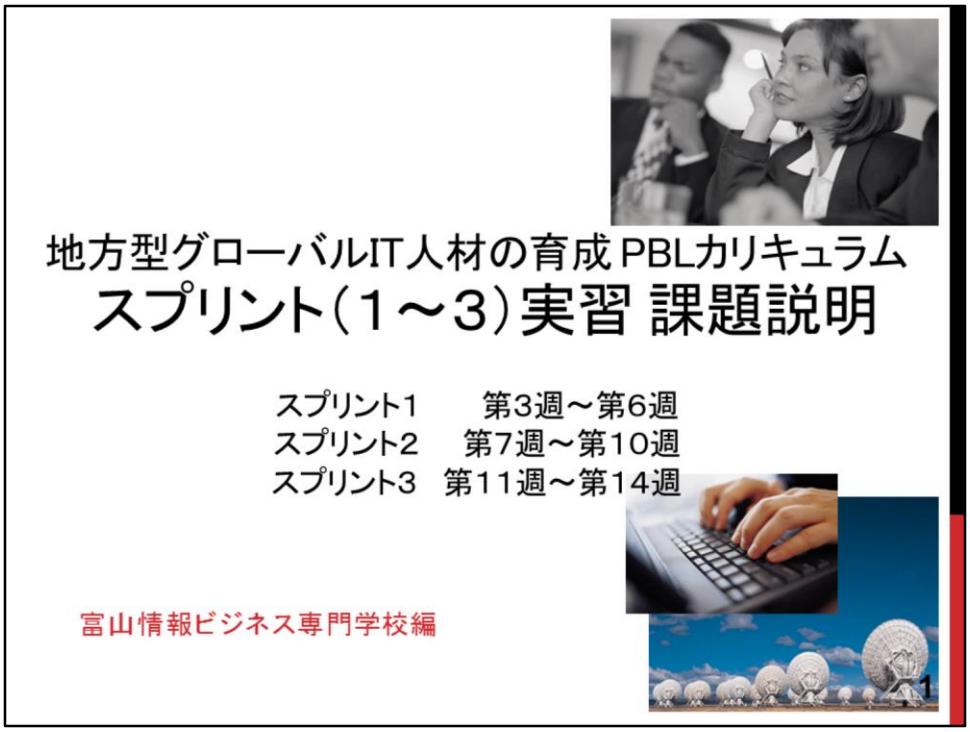
問２．スクラム開発手法を使ったプログラム開発はどのような手順で実施されるか？
（プロダクトバックログ、スプリントバックログ、デイリースクラム、
ふりかえりなどの単語を含めて説明しなさい）

問３．スクラム開発を行う際に特に気をつける必要がある事はどんな点か？

問４．開発環境の利用方法を理解できているか？ またサンプルソースの機能で理解できない点はどこか？

◯ここまでのアプリケーション開発準備がすべて完了した時点で、理解度の確認を兼ねて受講者にクイズを出して下さい。またこの時点でアプリケーション開発環境やスクラム開発手法に関する疑問点や感想を述べてもらい、不安要素を出来るだけ取り除いて下さい。実習を開始すると、毎日の作業に追われてゆっくりと考える時間はほとんど無くなります。

◯アプリケーション開発開発実習を開始する前に各チームで受講者それぞれが、どのような特技（得意な事や苦手なことなど）を持っているかを話しあう時間も設けて下さい。コミュニケーション術、交渉術、Java開発、Androidアプリケーション開発、PHPスクリプト、SQLデータベース操作、開発環境のメンテナンスなど得意分野をお互い事前に知っておくことで、開発作業がよりスムーズになります。



◯事前に説明しているスクラム開発手法の進め方に従って、スプリント1～3を実施します。各スプリント＝4週間ありますが、講師がスプリント中ずっと放置してしまうと、チームが崩壊する危険性が非常に高くなります。まずは、可能であれば毎回実施されるディリースクラムに参加して問題を抱えていないかどうかを客観的に判断します。もし問題点があるようであれば、ディリースクラム終了後に問題の担当者を集めて、解決方法を考えるミーティングを開催して下さい。ディリースクラムに毎回参加できない場合は、Redmine上に登録されるディリースクラム結果を確認し、問題点があるようであれば、別途当事者を集めて解決策の検討を実施して下さい。これらのミーティングを2～3回講師の主導で実施した後は、チーム内のメンバーのみで自発的に問題解決に向けた行動を起こせるようになります。

◯スプリントは全3回ありますが、スプリント1は開発環境やRedmine、スクラム開発手法に慣れる為のスプリントと割りきって下さい。アプリケーション開発実績よりも、開発環境に関する問題やスクラム開発手法の手順に関する疑問点を次のスプリントに引きづらないように気を配って下さい。スプリント1でリリースできなかった機能については、レトロスペクティブにてその原因を洗い出して、対応策をチームで決めるところまで行い、残機能をスプリント2に引き継ぐか、優先順位を変更して取りやめるかを決めておく必要があります。優先順位の変更はプロダクトオーナーのみが持っている権利なので、チームとプロダクトオーナーで決めるように指導して下さい。

◯スプリント2に入ると、ようやく当アプリケーション開発の難しさが見えてきます。Java言語、Google MAP API、PHP、データベース操作など取り組まなければならない問題が山積していることに気付きます。これらの問題点をチーム内で分担して解決に向かえる様に指導することが重要となります。この為、スプリント2でもやはりアプリケーション開発実績は、予定通り行かない可能性が高くなります。

◯スプリント3では、技術的な問題点がほぼ解消して、各メンバーは得意ジャンルで活躍できるようになります。スプリント3に入っても解決できない問題点は早めに切り捨てて解決可能な実装に目を向けさせる必要があります。また、このスプリントが最終なので、最終的にチームとしてどのような機能を確実に実装して完了させるかを明確にして、チームで完了に向けて全力で頑張る体制を取ることが重要となります。

進め方

<< 課題 >>
・スクラム開発手法でアンドロイドアプリケーション開発を実施する。
・１スプリント＝３週間で構成され、６時間/週 ｘ ４週 ＝２４時間＋時間外作業で開発を進める。
・元受会社側(日本側)は、オフショア側(海外側)と協働で進捗管理、機能設計、開発、質問や問題点の対応、Webサーバ・データベースサーバの保守運用、技術的課題の解決を主な作業とする。
・Redmineを通したオフショア会社側とのやり取りは全て英語にて実施する。
・ここでは１つのスプリント内で行われる作業を説明している。
・Redmineの使い方については“3.10_Redmine使用方法(BackLogs)”を参照。

<< 盛り込む内容 >>
①プロダクトバックログのメンテナンスと優先順位の決定
ユーザー要件を元に要望された機能をプロダクトバックログをRedmine上に登録する。また機能の優先順位も登録する。プロダクトバックログは、状況に応じて追加変更などを実施する。

②スプリントバックログの確認
オフショア会社側が選択し登録したスプリントバックログを確認して疑問や問題があったらRedmineを通して確認や変更を依頼する。

◯アプリケーション開発開発中は、常にRedmineに記録を残しながら作業を進める様に指導して下さい。些細な事でもRedmineに記録しておくことで、発表資料を作成する際の手助けとなります。

スプリント実施中で特に重要なのは、確実にディリースクラムを実行する事です。短い時間なので決められたスケジュールに沿って全員が参加することがスクラム開発を成功させる鍵になります。

海外メンバーとは、直接顔を合わせてミーティングすることは、ほとんど困難なのでSkypeなどによるTV会議システムを使用したり、SkypeやLINE、Facebookが持っているチャット機能を使ってミーティングを実施することをお薦めします。また、状況に応じてe-mailも使用します。Redmineによるコミュニケーションで不得意な点は、リアルタイムに会話をする事なので、この欠点を補うためのツールを予め決めて利用することを想定しておいて下さい。

進め方

<< 盛り込む内容 >>

③プロダクトオーナーは、設計された機能設計をRedmine上で確認し、疑問や問題点があれば確認し、問題が無ければ機能設計を承認する。承認された設計内容をチーム全体で確認し、全員が設計内容を把握しておく。

④発生した疑問や問題点の解決
プログラム開発中に発生した疑問や問題点をRedmineのチケットを通して受取り、問題解決を行う。また元受会社側で生じた疑問や問題点についてもRedmineのチケットとして、オフショア会社側に通知し、解決するように作業を進める。

⑤デイリースクラムの結果登録と確認
元受会社側及びオフショア会社側の参加者は、毎日ディリースクラムを行い、その結果をRedmineに登録する。疑問や問題点があれば時間を取って対応策を実施する。

⑥Webインターフェースやデータベースの変更が発生した場合、メンバー内のシステムアーキテクトに依頼される。システムアーキテクトは依頼内容を確認して、保守運用を実施する。また技術的な質問もシステムアーキテクトが答えるものとする。

◯既に実装されている機能の改善や新規機能の追加は、プロダクトオーナーとユーザとの間でまず話し合いが行われ、この内容をプロダクトオーナーが開発メンバーに伝える手順をとります。開発メンバーは直接ユーザと交渉しないように気をつけて下さい。プロダクトオーナーから受けた説明を元に開発メンバーは実装する機能のイメージを確定させてユーザストーリー化します。これにプロダクトオーナーは優先順位を付与して、プロダクトバックログとして開発対象に加えられます。ここで確定された機能の概要を取りまとめて文書化して、Redmineにアップロードしてメンバー全員がいつでも照会できるように整えておきます。

◯スクラム開発手法において、ディリースクラムは開発中毎日実施することが求められていますが、授業として当カリキュラムを実施する場合は、授業の進み方に合わせて、ディリースクラムをどのようなタイミングで実施するかを調整して下さい。より頻繁にディリースクラムを行うことで、チームの現状が細かく収集できるという利点が生まれますが、チームメンバーに負担がかかりすぎる場合もあります。状況を見ながら頻度を調整して下さい。但し、少なくとも1週間に1回以上は、ディリースクラム実施して下さい。

◯Webインターフェースやデータベースの変更が発生した場合、メンバー内にシステムアーキテクト作業ができるメンバーが居れば理想であるが、該当者が居ない場合は講師が変更を行い、その変更方法をチーム内のメンバーに教えることで、最終的にチーム内での自己運用できる事を目指して下さい。

進め方

<< 盛り込む内容 >>

⑦納品
プログラムの納品は、メンバー内のレビューが完了したものをプロダクトオーナーを中心として、検証して結果をフィードバックする。フィードバックする内容は、プログラムのバグや機能改善の依頼など。

⑧ふりかえり結果の確認
スプリント終了時にメンバー全員でスプリントのふりかえりを実施し、結果をRedmine上に登録する。この内容を確認して疑問点や依頼事項などを通知する。

⑨次スプリントへの準備
実装された機能とふりかえり結果を元にプロダクトバックログを更新し、次のスプリントに向けた準備作業を実施する。

⑩教師による各学生の理解度の評価と発表
評価方法に従って、各学生の作業状況を評価して、結果を発表する。

◯スプリントレビューはとても重要なミーティングとして位置づけられており、基本的にアプリケーション開発関係者全員が参加します。ここで実機によるデモを行い、動作に問題が無い事を確認できた機能のみをユーザへのリリース対象とします。場合によっては、リリース対象の機能が0件になることもあります。スプリントプランニングで予定された機能の実装とリリースが出来なかった場合は、その原因をレトロスペクティブ（ふりかえり）にて分析して、対応策を講じて次のスプリントで実施出来る様にして下さい。 完了しなかった作業については、残作業量を見積ってスプリントバックログを更新しておきます。残作業は、次のスプリントのプランニング時に継続するか取りやめるかを判断します。
また、レビューにおいて更に機能改善案や追加機能案が出た場合は、その案をチームで検討し、必要であればプロダクトバックログに加えて下さい。

◯スプリントの全作業が終了した時点で、講師が当スプリントにおける個々の受講者に対する評価を実施して、結果を発表します。講師と受講者は評価結果を元に、次のスプリントでどうすればより良く出来るかを話し合って対応先を考えます。

◯スクラム開発技法によるAndroidアプリケーション開発を一通り終了しました。これまで学習してきたことは、スクラム開発技法によるプロジェクト管理、Java言語によるAndroidアプリケーションの作成、PHP＋MySQLデータベースによるWebシステムの操作、チームによるソフトウェア開発、英語によるチーム内のコミュニケーションですが、一人では解決できない多くの困難に直面したと思います。これらの困難をチームの力で克服してこれたのではないかと思います。もし、チームメンバーがコミュニケーションを取らないで、自分に与えられた作業のみを行っていた場合、どのような結果になっていたでしょうか？ 想像してみてください。恐らくあまりにも多くの難しい問題に直面した為に、開発チームは空中分関して、ほとんどアプリケーション開発が進まなかったと思います。

◯チームによるソフトウェア開発は、一人では出来ないことを複数のメンバーにより、2倍3倍の効率で成し遂げる事が出来ます。この貴重な体験を今一度振り返って、成果発表をおこなって下さい。

◯成果発表の内容については、e-Learningでの述べているので、再度確認してみてください。また可能であれば英語での成果発表にチャレンジして見て下さい。

進め方

<< 課題 >>
・第１週目から１４週目までに学んだ内容を振返って、プレゼンテーション資料を作成して成果発表を行う。発表はチーム単位にまとめて行う。
・発表を聞く側の学生は、発表者に対して肯定的なフィードバックを必ず行う。
・教師からチーム毎に理解度に関する最終評価を発表して、総括を述べる。
・基本として元受会社側（日本側）学生を発表対象とする。

<< 盛り込む内容 >>
①最終発表のプレゼンテーション資料の内容は下記の通りとする。
(1)スプリント毎に実施した作業は何で、その役割を経験して、どのような結果となったか？　また難しかった点と得られた経験は何か？
更に自分なりに工夫した点は何か？
(2)もう一度同じ作業を行うとした場合、事前にどのような準備をして、どのような事を考慮して問題に取り組むか？
(3)どのような問題点が発生して、あなたはそれに対してどのような対策を提案して実行したか？ どのような結果となったか？
(4)今回使用したプロジェクト管理ツールが使えなかったとしたら、どのようにしてオフショア会社側を管理するかを答えなさい。
(5)これまで経験したシステム開発とアジャイル開発（スクラム）では、何が異なっているか？ アジャイル開発の長所と短所は何か？

◯これまでのアプリケーション開発の歩みをスプリント1からスプリント3まで順を追って説明して下さい。特に直面した大きな問題点について、どのような問題が発生して、どのように解決したかを詳細に発表して下さい。またウォータフォール開発手法によるチーム開発とスクラム開発手法によるチーム開発でどのような違いがあったかについても述べて下さい。

◯外国人とのコミュニケーションについて、述べて下さい。また今回一緒に開発に参加した外国人の特徴を述べて下さい。もし日本人だけでチームを作った場合、今回のチームによる開発とどのように異なった結果になったかを想像して述べて下さい。

◯今回経験したスクラム開発手法は、どのようなソフトウェア開発に向いているか意見を述べて下さい。

進め方

＜＜　盛り込む内容　＞＞

②成果発表は、開発チーム毎に実施し、発表者１人あたり５分間とし、チーム全体の発表が終了した後に質疑応答を５分間実施する。

③発表チーム以外の学生が発表者に対してフィードバックを行う。

④チーム毎に教師から各学生の理解度に対する最終評価の発表する。

⑤全チームに対して発表を繰り返す。

⑥教師が実習全体を通しての総括をする。

◯成果発表は、チーム毎の発表になります。プロダクトオーナーがチームの発表を司会者として取りまとめて下さい。発表順や発表方法などについてもプロダクトオーナーが中心となって決定し、発表を行って下さい。

◯全チームの成果発表が終了した後に、講師から総括を述べます。総括には、各チーム毎の優れていた点、もっと改善したほうが良い点を述べ、チーム内のコミュニケーションの良し悪しについて述べて下さい。また開発したアプリケーションに関する評価も述べて下さい。

◯最後に講師から個々の受講生に最終的な評価を100点満点で述べて下さい。この時点で受講生が納得するまで評価内容を講師及び受講生で突き詰めて下さい。

◯マスタースケジュールは、授業時間数の長短に合わせて適宜調整して下さい。1スプリント＝4週間、全3スプリントで構成されていますが、1スプリント＝2週間～4週間の期間で調整することが出来ます。また全3スプリントに拘る必要は無く、全2スプリント～6スプリントとしてマスタースケジュールを調整する事が出来ます。スケジュールは、受講者のスキルレベルに合わせてください。

◯サンプルソースとして付属しているAndroidアプリケーション開発は、かなり難易度の高いソフトウェア開発になっています。こちらについても受講者のスキルレベルに合わせて、開発内容を適宜置換えて下さい。

スクラム入門（基礎編）

Scrum Simulation
（with LEGO® Bricks）

2015年9月7日 初版

富山情報ビジネス専門学校 編

当講座は、スクラム入門及びスクラムシミュレーションのどちらもグループによる作業が発生します。
その為、1グループ＝3～6名程度となる様に、机と椅子を準備し、予めグループを作って受講に備えて下さい。
どちらの講義も最初に決めた同じグループで作業を行います。

過去の実績から1グループ＝6名が最もバランスが取れており、進めやすいと思います。

またスクラムシミュレーションの準備として、文房具を各テーブルに準備して下さい。レゴブロックを使える場合は、レゴブロックを広げて置く為の机を別途準備します。

謝辞

当資料は、これからスクラムを学ぼうとしているすべての人の為に作成されました。また“The Scrum Guide”（著者：Ken Schwaber and Jeff Sutherland) を参考にして作成されています。

スクラムの一歩は、まず上記のドキュメントを読む事から始まります。

このような素晴らしいドキュメントを公開された著者及び運営団体に感謝します。

※ 上記公開先URL: http://www.scrumguides.org/scrum-guide.html

※ 日本語翻訳版: http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-JA.pdf

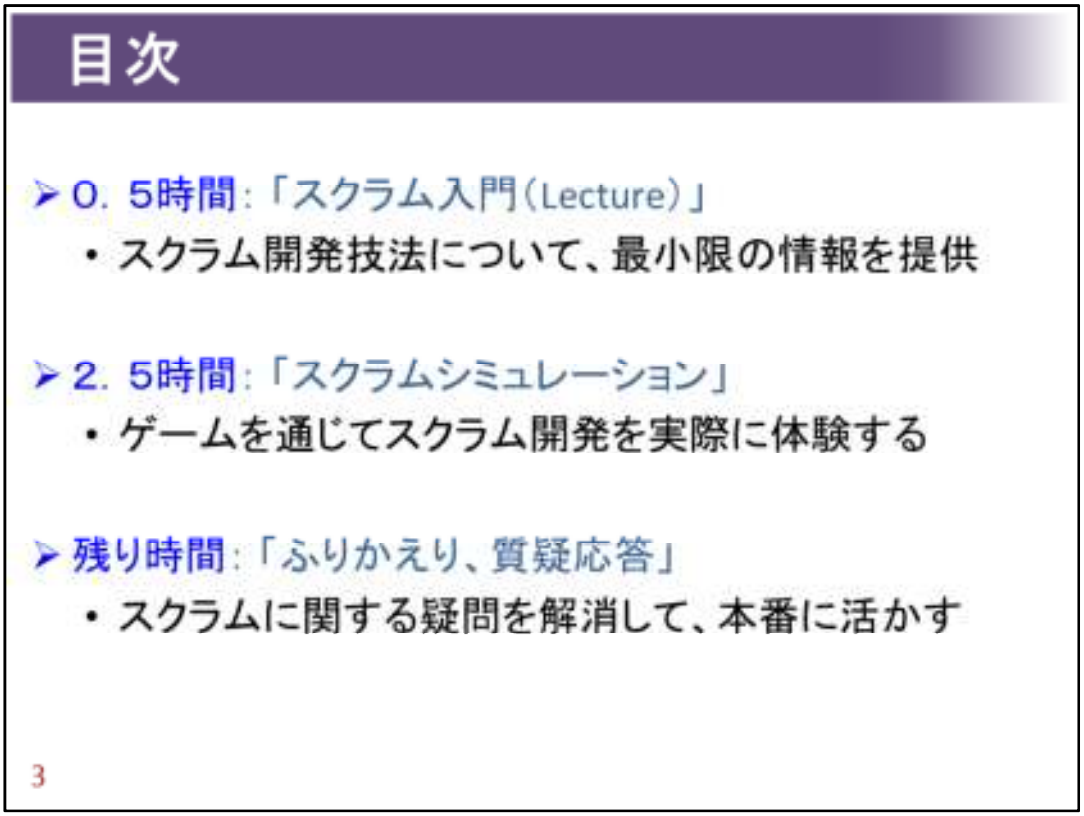
（日本語版は、ワイクル株式会社 角 征典(かどまさのり）氏によって作成されました。）

また当教材を作成する前に角 征典先生の『LEGOではじめるスクラム入門』を受講させて頂き、非常に感銘を受け、影響を受けました。もしレゴスクラムに興味を持たれたらワイクル株式会社角 征典先生の当該講座を受講されることをお薦めします。

※ URI: http://www.waicrew.com/training/scrum/

2

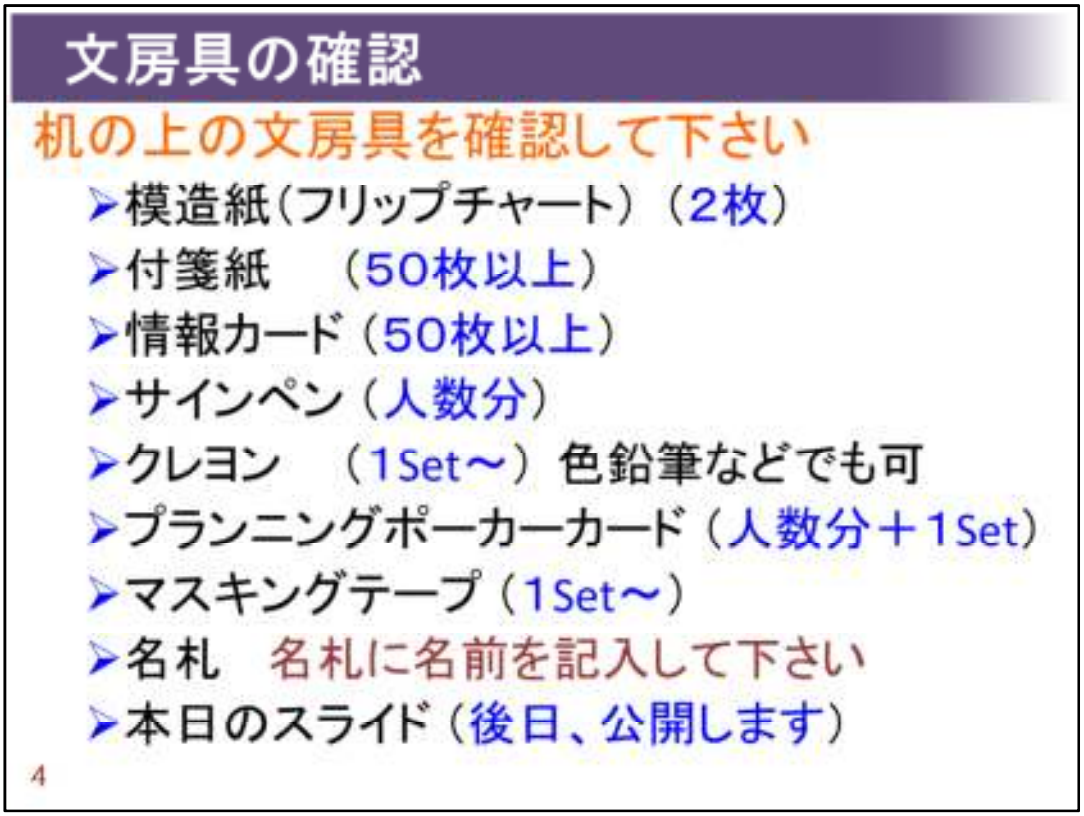
当スクラムシミュレーション教材は、ワイクル株式会社 角氏に相談し、得られた情報が元になっています。当ドキュメントを作成する上で角氏には非常にお世話になりました。



最初のスクラム入門は、座学です。
ここでスクラムの基本を一通り知識として頭に入れて、次のスクラムシミュレーションを実施できる基礎を学びます。

スクラムシミュレーションでは、スクラム開発技法を1から順番に体験を通して学びます。楽しく学ぶことを第一の目的とします。2．5時間は長いように思われますが、時間に追われながらの作業になり、あっという間に終わります。

最後にスクラム入門全体をふりかえります。
ここで今回のシミュレーションには含まれなかった作業などを補足します。



スクラム入門及び、スクラムシミュレーションを実施する為に、上記の道具を揃える必要があります。
これ以外に、レゴブロックがあれば更に完璧です。
レゴブロックが無くても、情報カード（3inch x 5inchの厚紙）があれば、これに絵を描いて立てたり、複数枚を貼り合わせて立体的な建物にしたりして使うことにより、ブロックの代用品にする事が出来ます。机の上で街作りを盛り上げる道具があれば何でも利用可能です。例えばiPadに動画を表示して配置する事により、野外映画館として使ったりというアイデアも大歓迎です。その他、小さな人形やぬいぐるみなどがあっても盛り上がります。

最初のスクラム

1人の日本人と2人の偉人

◎日本の製造業のイノベーションプロセスとして、「スクラム」を世界に紹介した日本人
⇒ 一橋大学大学院名誉教授 野中郁次郎博士

◎アジャイル開発技法としてソフトウェア開発向けの「スクラム」開発技法として育てた
アメリカの2人

1995年「SCRUM Development Process」にて公開

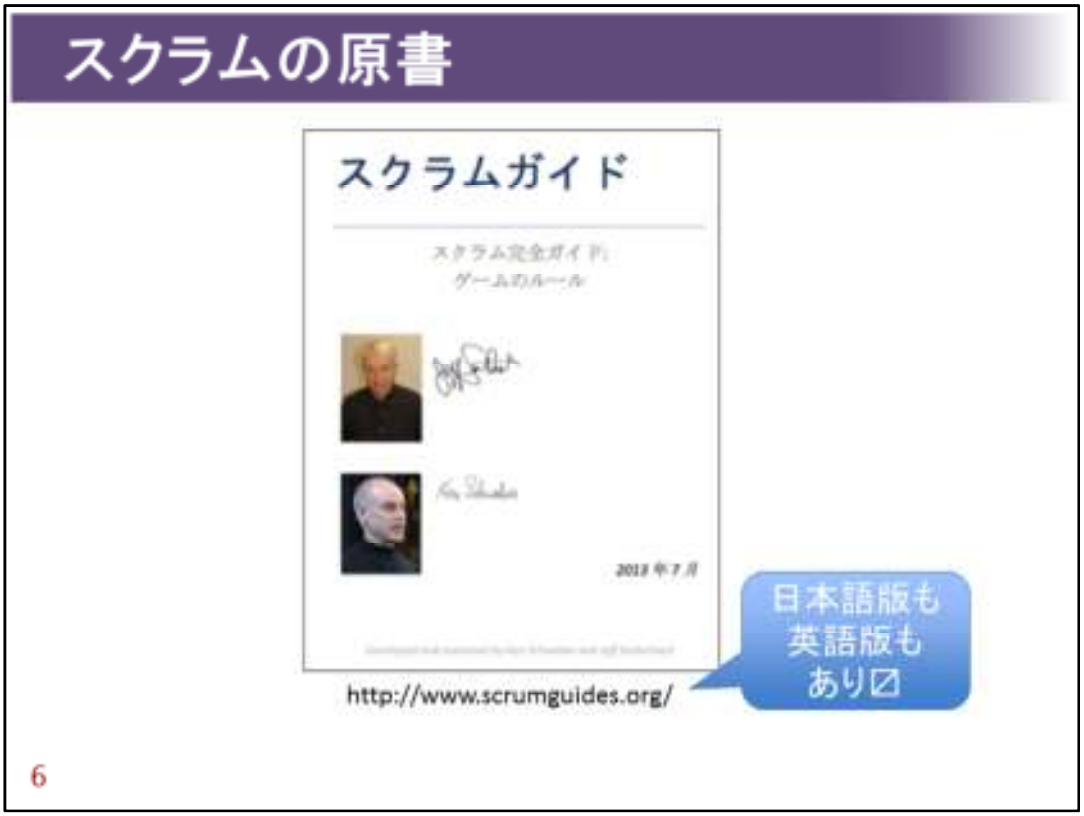
野中郁次郎博士

ジェフ・サザーランド博士

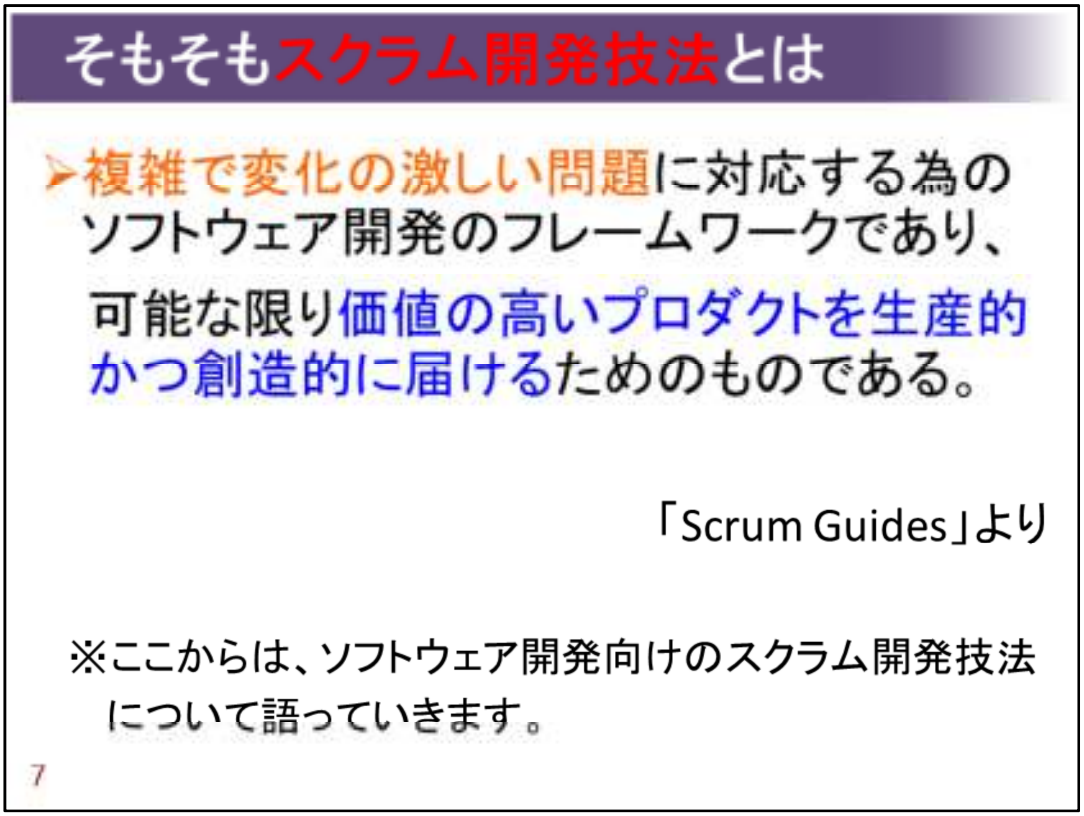
ケン・シュエイバー氏

5

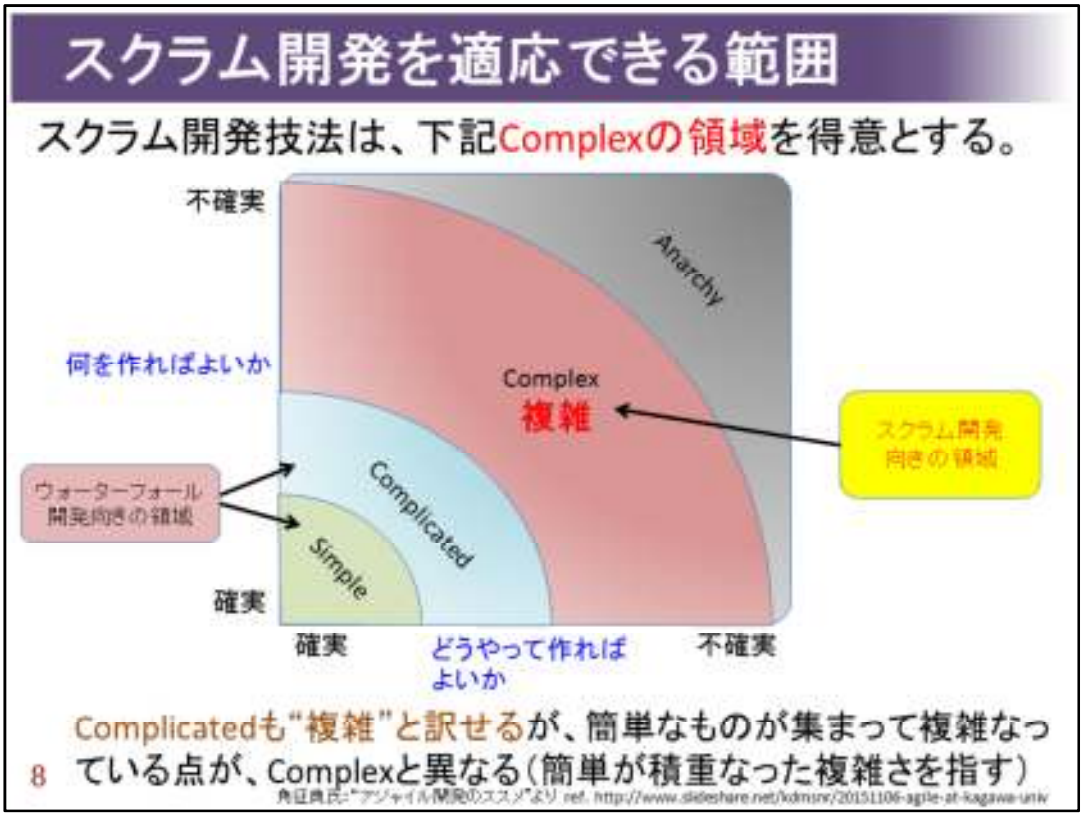
スクラムの原型は、日本のトヨタ自動車などの製品開発手法で、これを野中博士の論文により世界に知らしめる事になりました。これをソフトウェア開発分野に最適化させたのが、サザーランド博士とシュウエイバー氏です。



インターネット上にスクラムの基本学習書として、「スクラムガイド」がフリーで配布されています。スクラムに携わる人々は、まずこの教科書に目を通すことから始めます。
もちろん英語版の他に、日本語版もあります。この日本語版翻訳は、前出のワイクル株式会社角氏によるものです。
上記以外にもスクラム入門用の優れたドキュメントがインターネット上にいろいろとあるので、探してみてください。
但し、中には古いスクラム仕様のドキュメントもあるので、「スクラムガイド」と内容の照合を行ってスクラムのすべてを最新版として網羅しているかを確認しておく必要があります。

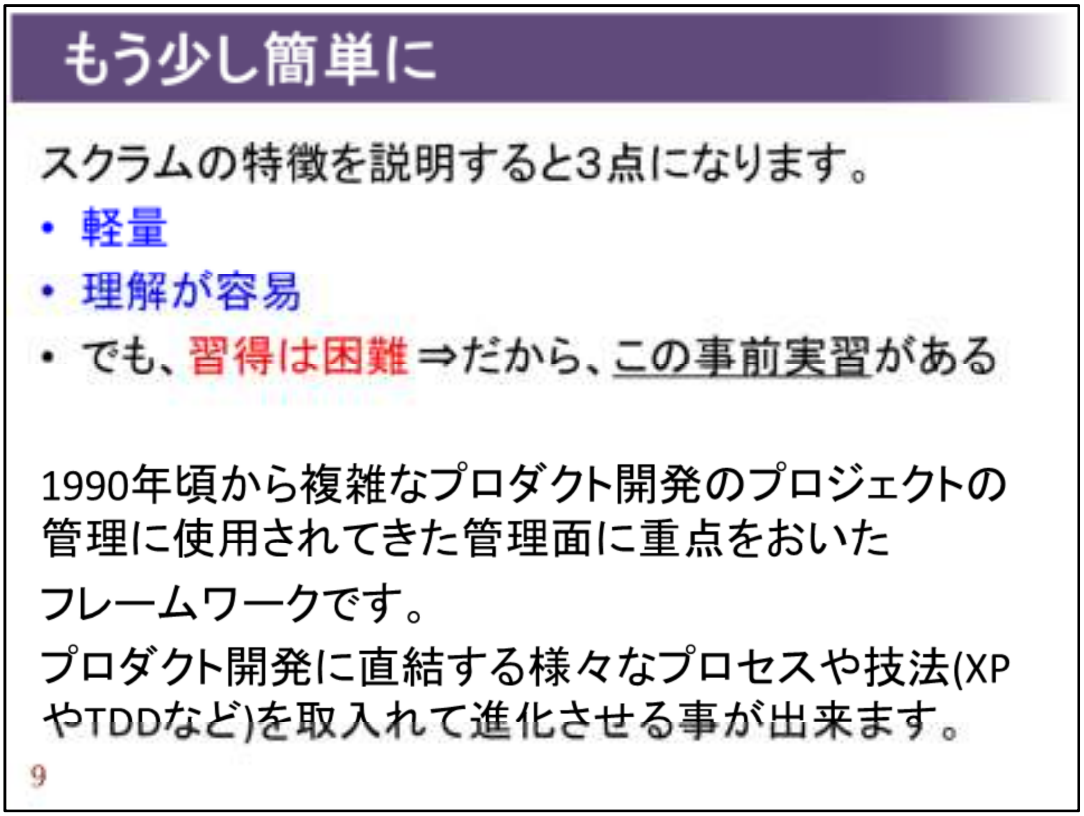


内容の通り、簡単すぎる開発には向きません。
逆に少し難しいと考えられる開発に適用する事で、期待以上の良い結果を得る可能性が高い開発手法と言えます。



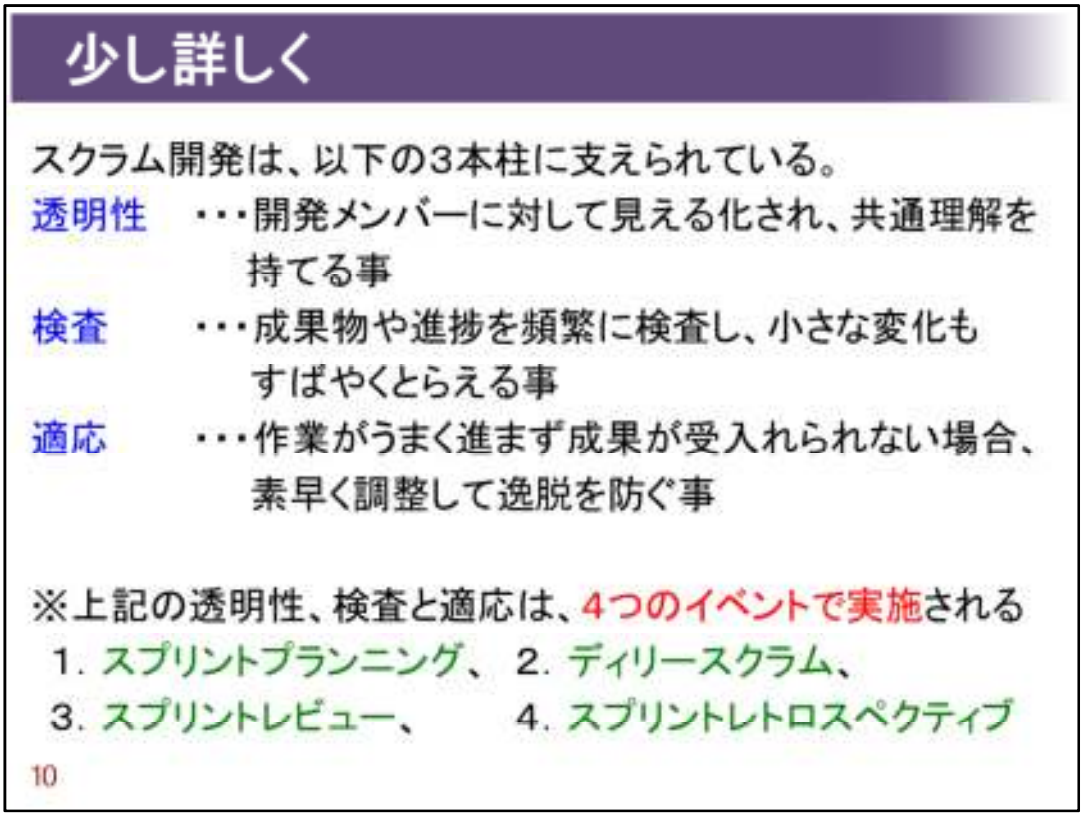
ウォータフォール型の開発であれば、開発の手法（どうやって作れば良いか）が既に明確になっており、基本的に何を作れば良いかに注力して開発を進めれば良いと考えて下さい。スクラム型の開発手法においては、何を作れば良いかと、どうやって作れば良いかについても開発プロセス内に取り込んで進化させていく形になります。

スクラムは、“もしかして実現困難かも・・・”と考えられる様なシステム開発に向いています。これはユーザの代表者をチームに加えて、開発対象スコープを随時調整しながら開発を進める為、顧客満足度が高い成果物となる可能性が高い手法です。
Complicatedとは、一見難しそうに見えるが、実は簡単なもの(機能)を大量に組み合わせている為に複雑化しているシステムを指します。細かく機能を分解していくと、実は簡単な機能の集合体となっており、コツコツと開発を積み上げることにより、実現できるようなシステムで遠くにあってもゴールはしっかりと見えているシステム開発を想像して下さい。
スクラム開発が適しているComplexとは、Complicatedの様に分解できず、機能の実装方法がすぐに見当できないようなシステム開発を指します。その為、チーム内にユーザの意見を常に取入れて開発の方向性を調整しながら開発を進める事になります。



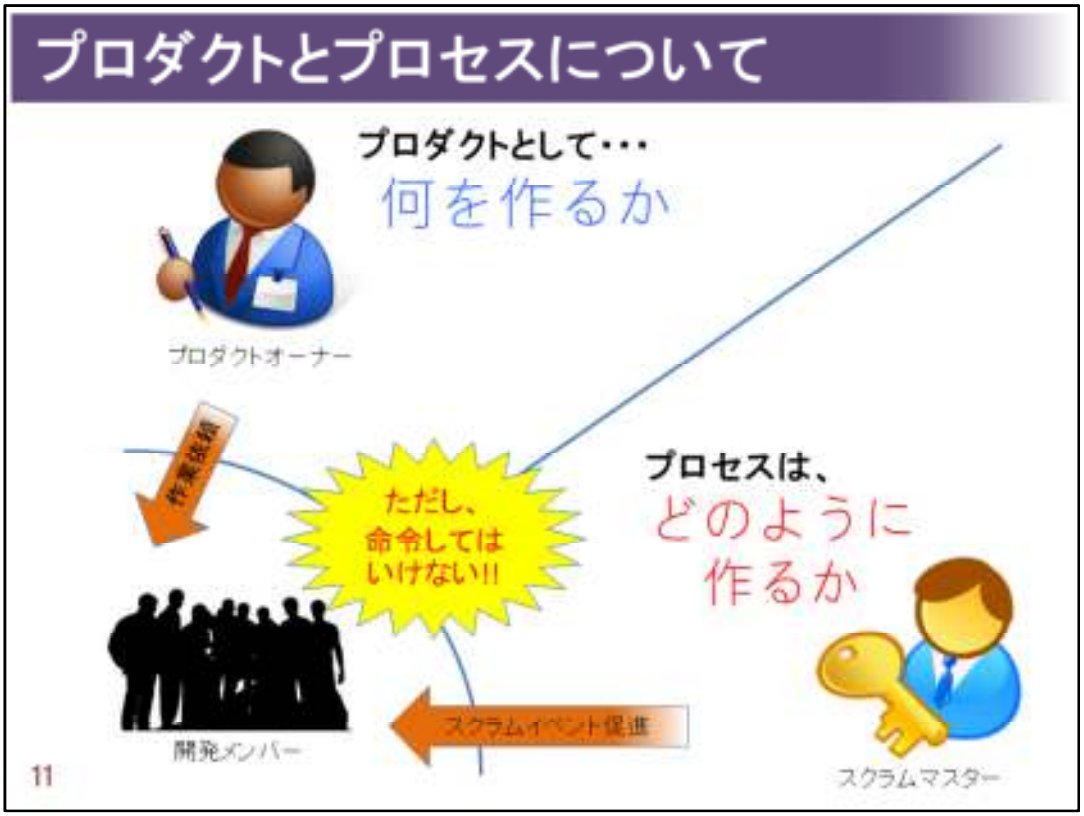
スクラム開発手法が簡単と言える根拠は、手法としての決め事が比較的少なく分かりやすいので、“軽量”であり“理解が容易”と言えます。
しかし、軽量の裏返しで細かな点に関しては、実行者の裁量に委ねる事が多く、知識としてスクラム開発手法を習得しただけでは、実行するイメージが湧きにくい欠点もあります。
その為、何回かスクラム開発を経験した経験者をチームに含めて、チーム全体の開発力を高めるて行くのが本来のやり方ですが、授業においては全員がほとんどスクラム開発の経験が無い為、スクラムシミュレーションを体験することにより、習得の困難さを解決しようとする試みとなっています。

また、スクラム開発手法は、プロジェクト管理側面を得意とし、実際の開発に関する技法等については、実行者任せとなっています。実際の開発においては、XPのペアプログラミングやTDD（テスト駆動型開発）を実際の開発手法としてチームで利用することを決定し、開発を進めていく必要があります。
スクラムをはじめとするアジャイル開発手法の場合、何回でも同じプログラムコードを変更・追加し、機能拡張する為、自動テストを併用する事を強く推奨します。これまでの開発で自動テストを実施していない場合は、学習する時間を割いて自動テストを習得する機会を設けることをお薦めします。

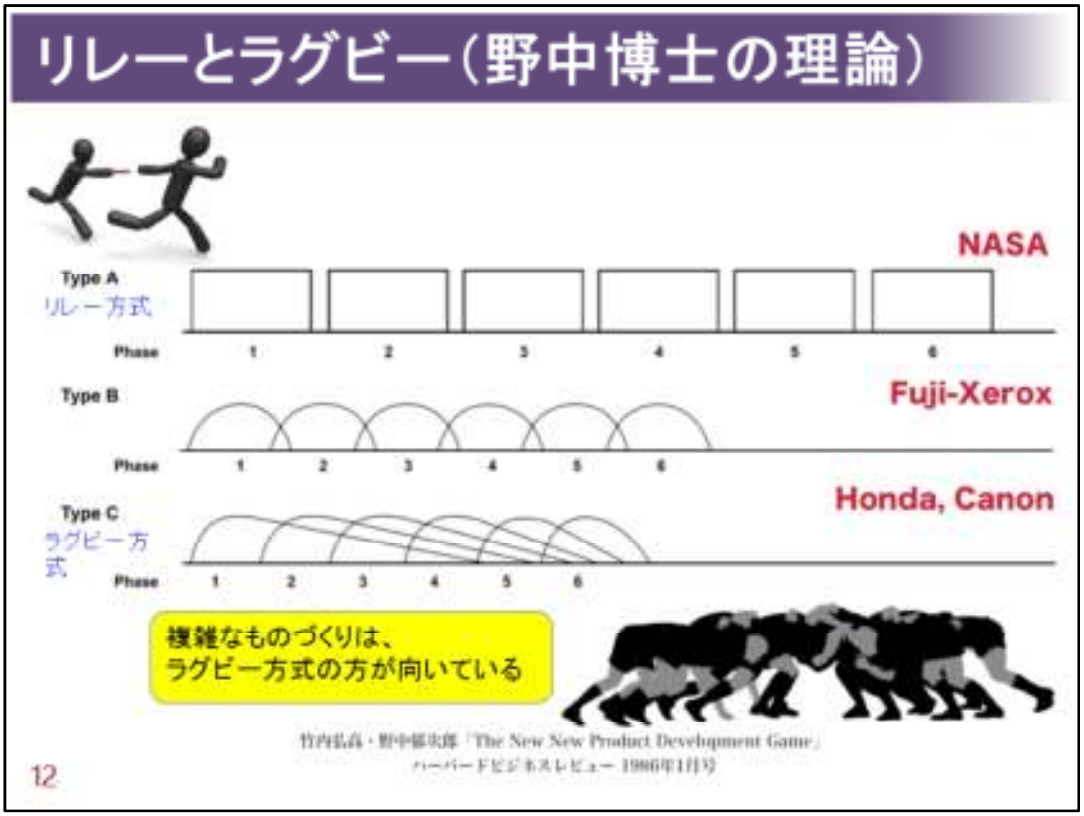


スクラム開発の根本に“3つの柱と4つのイベント”という考え方があり、これに沿っていればスクラム開発から外れる危険性が減ります。
“透明性”とはスクラム内で実施しているすべての作業について、いつでも誰でも確認することが出来る環境を作って作業を進める事を指します。各レビュー結果も見える化します。
“検査”とは、短いサイクルで、開発対象の選定・開発・レビューを繰り返すことで、何回も検査を実施したシステムをユーザに提供できる強みを指します。
“適応”とは、開発手法が仕様変更に強く、常にユーザ目線で今必要な機能の洗い出しと実装を容易に取り込むことが出来て、方向性の修正も随時可能な柔軟性を指します。

“スプリントプランニング”は、今回の開発サイクル（スプリント）内で、対象とする機能を選択し、作業量を見積って作業タスク化する作業を指します。この作業から各スプリントが開始されます。
“ディリースクラム”は、朝会とも呼ばれ、毎日実施する進捗報告会で、基本は一人1分以内の報告で実施します。ここで報告された内容に関する対策は、ディリースクラムが終わった後に有識者が集まり、打合せを実施します。
“スプリントレビュー”は、開発サイクル内で開発され実装された機能をユーザも含めてレビューし、ユーザへのリリースを行うか、やめるかを判断します。
“スプリントレトロスペクティブ”、レトロスペクティブとは“ふりかえり”という意味で、スプリントの最後に、今回のスプリントを振り返って、次のスプリントに向けた改善点を協議し、適用します。



スクラム開発では、プロダクトオーナーから開発メンバーに“何を作るか”(プロダクト)の指示を出します。また、スクラムマスターは、開発メンバーを緩やかにマネジメントします。また、“どのように作るか”(プロセス)をメンバーと共に考えて作業方法を改善していきます。
開発メンバーに対して命令すると、以前までのプロジェクト組織とかわらなくなってしまいます。あくまでもチームの意思決定を促し、チームとして考えて実行します。

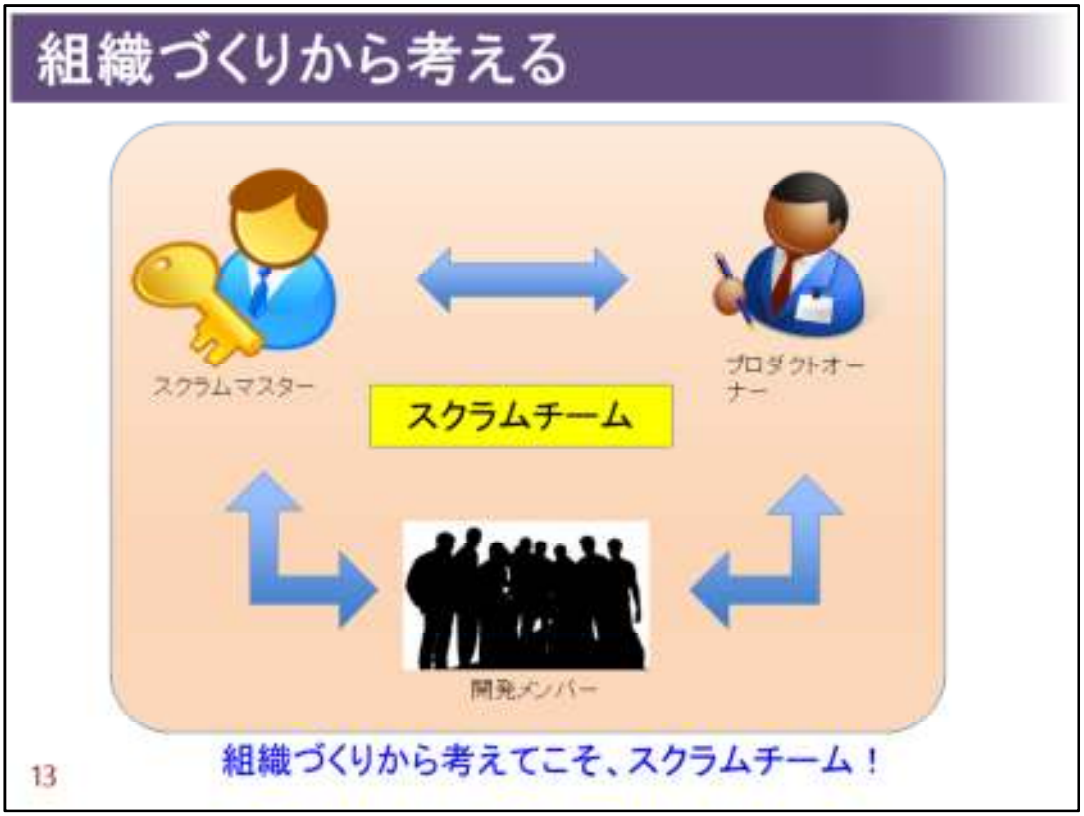


そもそもソフトウェア開発は、TypeAの様に分業式ではじまりました。1の作業を終わった後に、2の作業を実施し、3の作業を・・・・
というように進んでいきます。この方式をリレー方式と呼びます。これは、NASAによるロケット開発などにも利用されている開発手法で、担当者の作業範囲を明確に細かく分けて管理するアメリカ式の方法と言えます。
非常に効率的で良いのですが、もし前工程までに問題が発生した場合、後戻り作業が多く発生し、スケジュールの見直しが必要になります。
TypeBは、富士ゼロックスで行われていた方式で、各作業工程の最初と最後を重ねることで、工程の引き継ぎと後戻りの危険性を減らしています。
TypeCでは、更に作業工程を重ねる事で、常にすべての工程を改善し進化させる事ができると共にチームメンバーが一体となって作業を実施することが求められます。
この為、TypeCは日本的な手法で、HondaやCanon、Toyotaなどの製品開発に適用されていました。この方式をラグビー方式と呼びます。

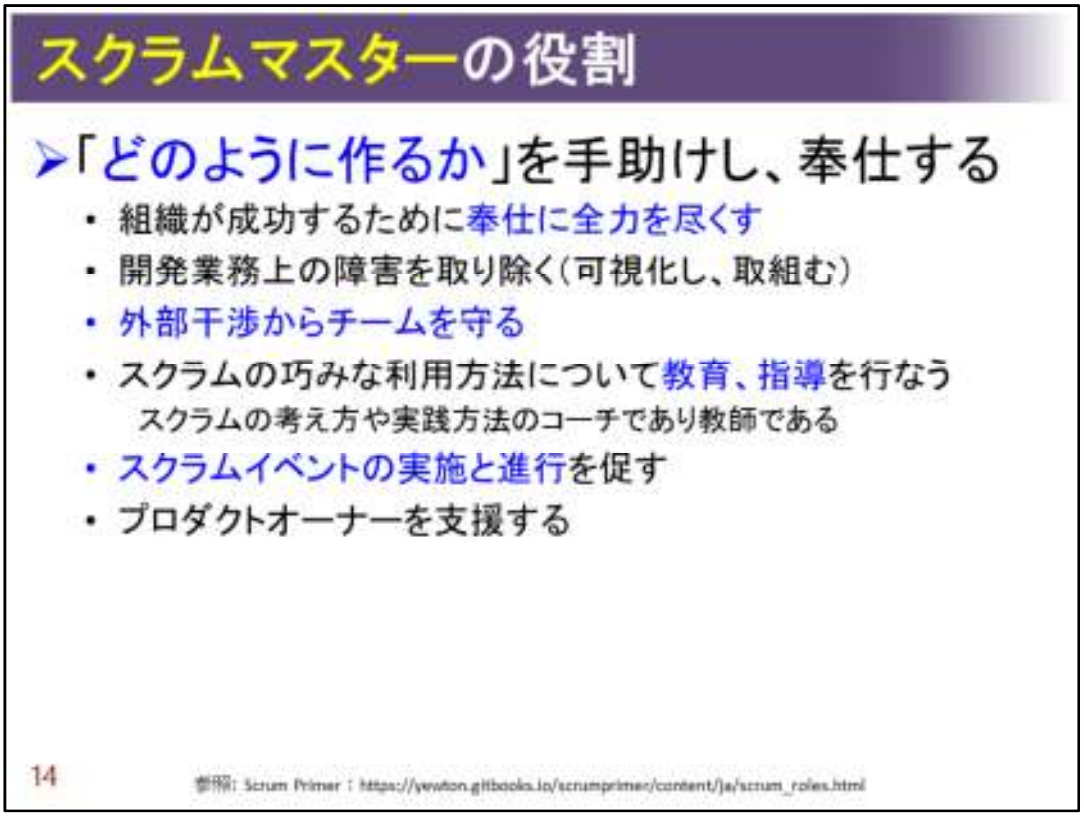
また、ラグビー方式には、下記1～6の特徴があります。
1. 不安定な状態
2. 自己組織化
3. 開発フェーズの重複
4. マルチ学習
5. やわらかなマネジメント
6. 学びの共有

難しそうな事を言っていますが、チームに責任と権限を持たせて、チーム一丸となって開発に取り組み、チーム全体のパフォーマンスを上げていくのに適した方式です。
開発対象をチーム内で管理できる位の細かな単位に分解して、設計ー開発ーリリース

を繰り返します。作業の分担をあまり明確にしないで、誰でもトライできる環境になっています。誰でもトライして成長できるのでマルチ学習と呼ばれます。マネジメントとしては最小限の干渉しか行わず、チームの自立性に任せます。
各メンバーが得意な技術力を発揮し、チームメンバーと一緒に作業することで、常に他者からの刺激と学びがあります。



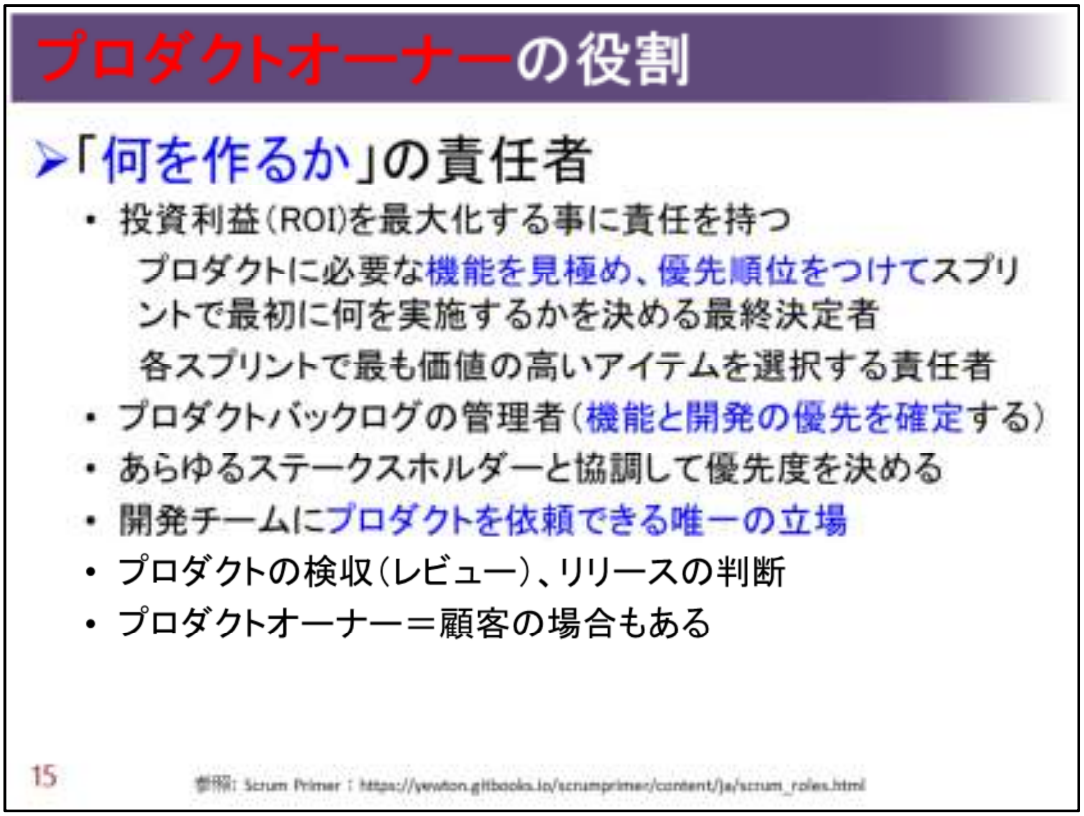
これらの組織を作るために、スクラムチームを構成します。
プロダクトオーナー（1名）
スクラムマスター（1名）
開発チーム（6名 ＋ー3名）　6名が理想で、3名～9名で開発チームを構成出来る様にする
の組織をつくります。



わかりやすい言い方をすれば、スクラムチームのマネージャーです。但し、作業指示やとりまとめは行いません。
開発における中心は、あくまでもプロダクトオーナーとチームメンバーです。チームリーダでもチームの代表者でもなく、チームに奉仕する役割。
あくまでも開発チームが自発的に動きやすくする為も環境づくりに徹します。

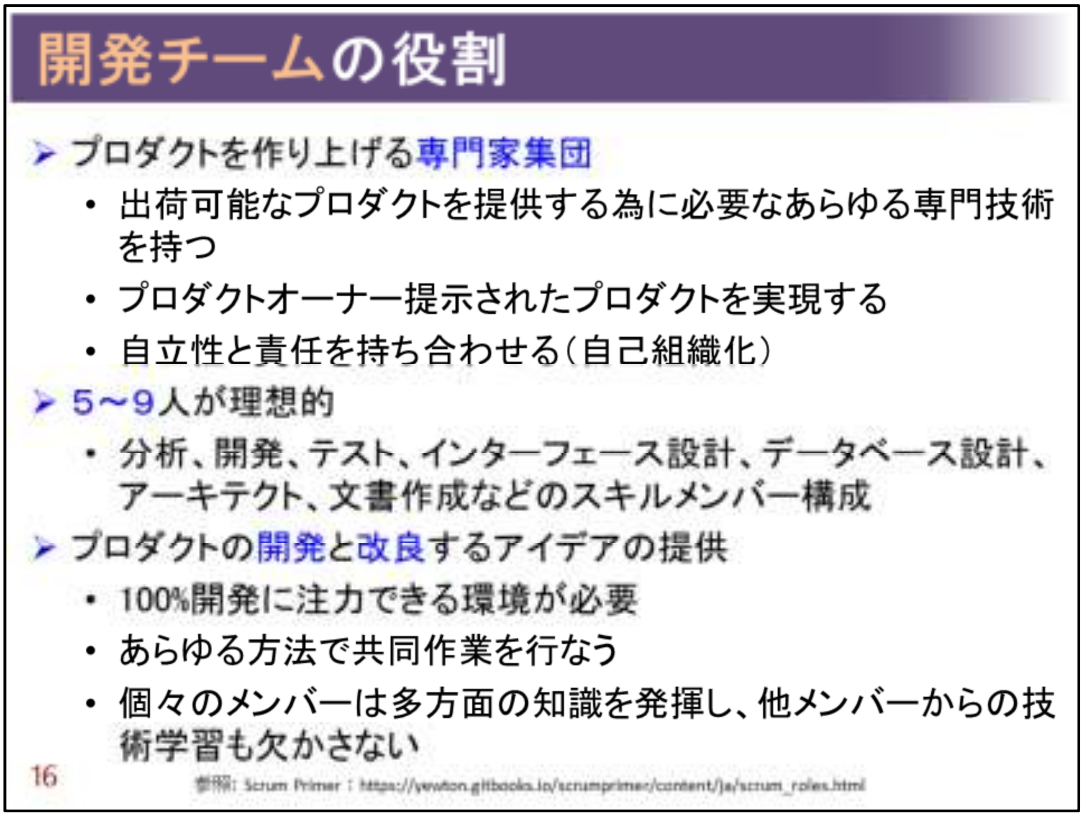
チーム内に問題が発生した場合、メンバーと一緒に解決策を考えます。
チームの外部からチームを守ります。メンバーが開発に集中できるようにします。
4つのイベントを通知し、進行する様に促します。

スプリントプランニング、ディリースクラム、スプリントレビュー、スプリントレトロスペクティブに参加します。



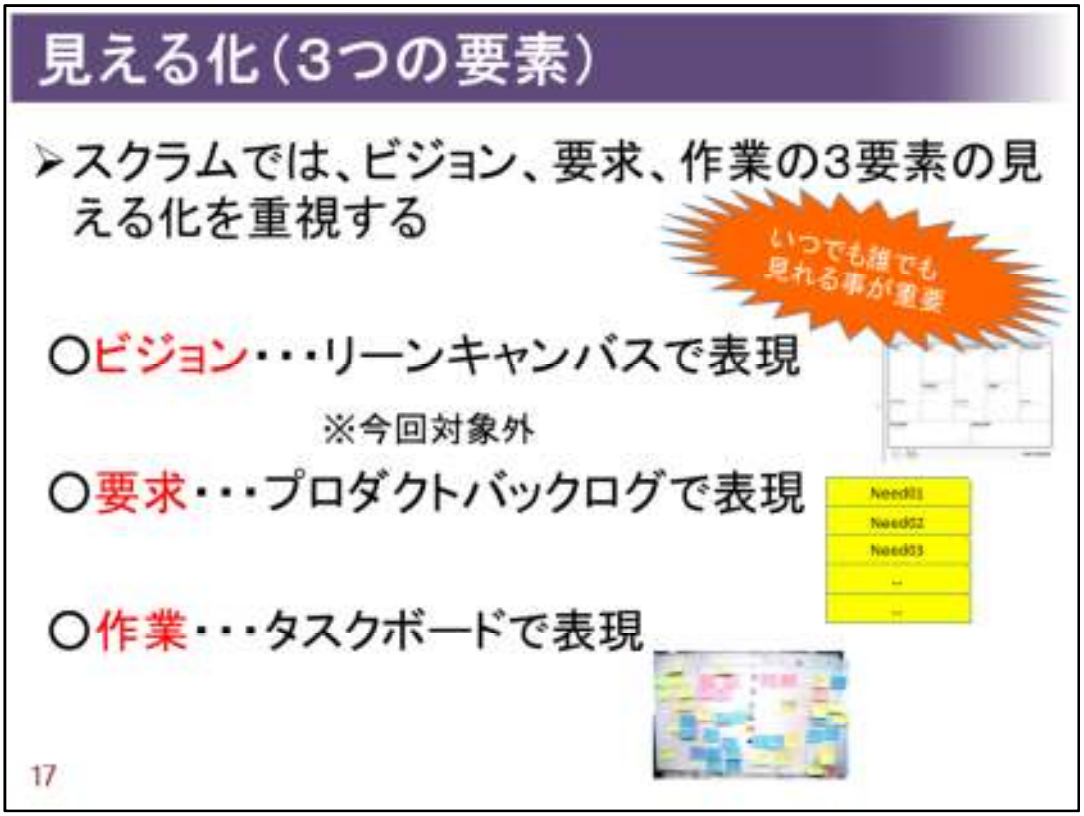
プロダクトオーナーは、ユーザーの代弁者です。
実際の開発では、ユーザ側から選出されてスクラムチームに加わる事もあります。(この方が良い)
ユーザがやりたいことをユーザの言葉で開発チームに伝えます。その為、技術面の知識は無くても良い。
開発チームは、プロダクトオーナーに機能や仕様を確認し、仕様を確定します。
プロダクトオーナー以外は、仕様に関する決定権はありません。
作成されば成果物のレビューと、ユーザへのリリースの判断を行います。

一番重要な作業は、後述のプロダクトバックログを作成し、管理することです。プロダクトバックログとは、実装したい機能とその優先順位が記録されたリストです。
プロダクトオーナー以外は、プロダクトバックログを変更する事は出来ません。
また、スプリントプランニング及び、スプリントレビューに参加します。スプリントレトロスペクティブにも参加することが出来ます。



開発チームは、実際にシステム開発を行うメンバーの集まりで、細かな作業分担は意識的にしません。
常に高いモチベーションを持って、自分からチームの為に出来ることを率先して行います。

開発チームの行う主な作業は下記の通りです。
スプリントプランニングの参加
スプリントバックログ（後述）の管理
作業タスクの作成と作業実施
プログラム開発の実施
作成したプログラムの単体テスト
開発済み機能のレビュー
ディリースクラムへの参加
スプリントレビューの参加
リリース作業
スプリントレトロスペクティブの参加



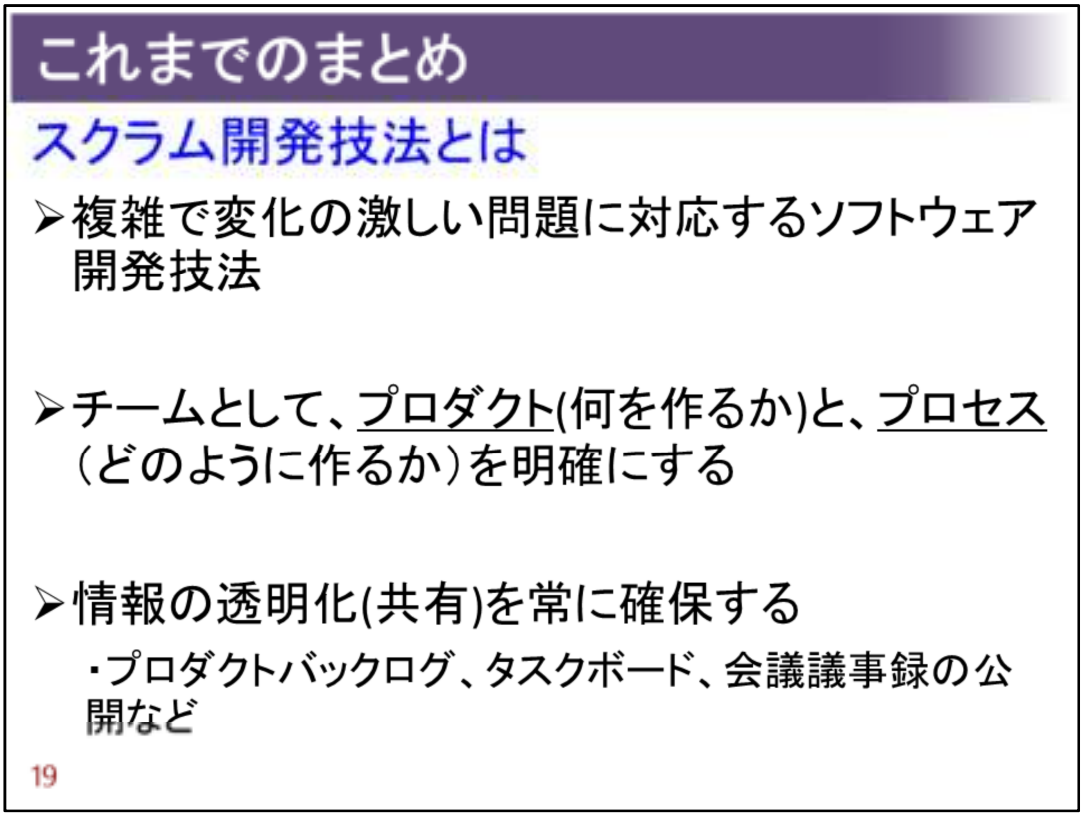
システムの全体像が分かるようにリーンキャンバスを利用します。（今回は対象外）
システムにどのような機能が必要かまた優先順位はどうかを表すプロダクトバックログを利用します。
スプリント内の作業タスクの進捗を管理するためにタスクボードを利用します。

これらは、常に開発メンバーが確認できる所に配置され、いつでも誰でも確認出来る様に準備します。

スクラムの3要素に加えて、実施される会議も見える化します。特に会議のスケジュールの見える化が大切です。

今回の教材で行う実際の開発では、1スプリント＝4週間とします。また、全部で3スプリント実施します。
各スプリントは、スプリントプランニングからはじまり、レトロスペクティブと製品のリリースで終了します。

どのように行うかは、スクラムシミュレーションで詳しく説明します。



これで、スクラムの概要説明を終了します。
ここまでをまとめると、このようになります。

スクラムは、どのようなシステム開発に向いているか？　複雑で変化の激しい問題を含んだ開発

常に検査（レビュー）し、現状に適応して行きます。
その為に、2つのフィードバックを有効に働かせます。“何を作るか”“どのように作るか”

スクラムチームは、自己組織化されたチームです。
プロダクトオーナー、スクラムマスター、開発チームのそれぞれの役割はどのようなものか？
情報の透明性（見える化）の為にどのような工夫をしているか？
プロダクトバックログ
タスクボード（カンバン）
各会議の議事録の保管場所

基礎編 終わり
20

スクラム入門(実習編)

Scrum Simulation

(with LEGO® Bricks)

2015年9月7日 初版

富山情報ビジネス専門学校 編

スクラムシミュレーションの準備をして下さい。
グループ毎に、それぞれの机の上に模造紙を１枚貼ります。また近くの壁もしくはホワイトボードにも模造紙を１枚貼ります。（各グループ２枚の模造紙を使います）
机の上に模造紙を貼る時は、どちらか一方の面に合わせて模造紙を貼り、反対側に空きスペースを設けて下さい。

レゴブロックを使える場合は、レゴブロックを広げておける机を別途用意して、ブロックを広げて下さい。

謝辞

当資料は、これからスクラムを学ぼうとしているすべての人の為に作成されました。また“Scrum Simulation with LEGO Bricks”（著者：Alexey Krivisky, Creative Commons Attribuion 3.0 Unported License) を参考にして作成されています。

このような素晴らしいドキュメントを公開された著者及び運営団体に感謝します。

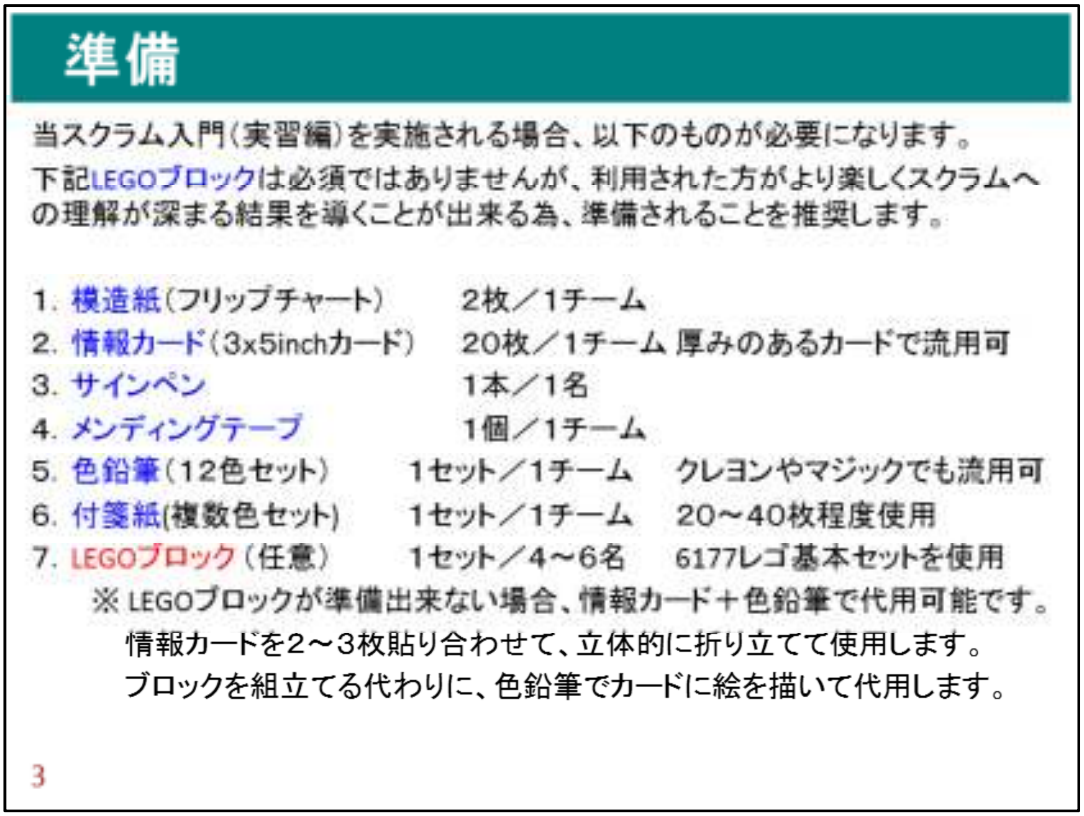
※ 上記公開先URI: http://www.lego4scrum.com/p/scrum-with-lego.html
※ LEGOのサイト: http://shop.lego.com/en-US/LEGO-Basic-Bricks-Deluxe-6177

また当教材を作成する前に角征典先生の『LEGOではじめるスクラム入門』を受講させて頂き、非常に感銘を受け、影響を受けました。もしレゴスクラムに興味を持たれたらワイクル株式会社角 征典先生の当該講座を受講されることをお薦めします。

※ URI: http://www.waicrew.com/training/scrum/

2

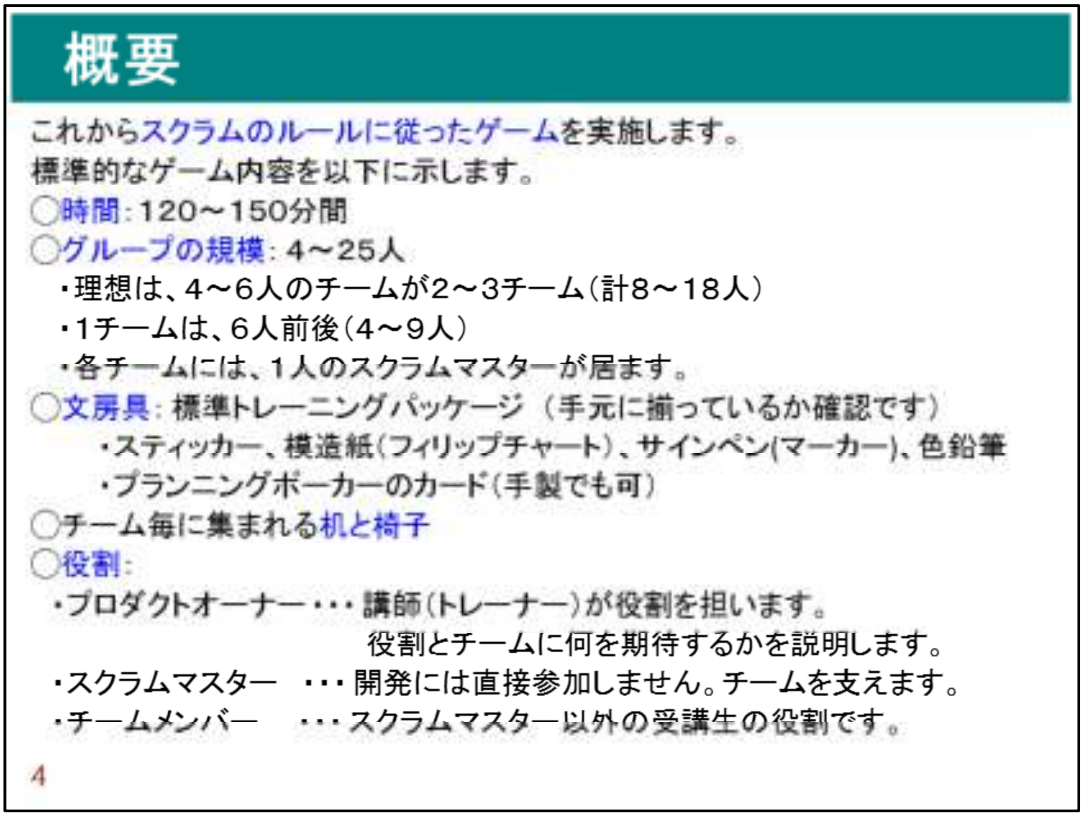
当教材は、“Scrum Simulation with LEGO Bricks”をベースに改良を加えています。このアイディアは、ワイクル株式会社 角氏から頂きました。角氏は、別途有料のレゴスクラム講座を開催されています。



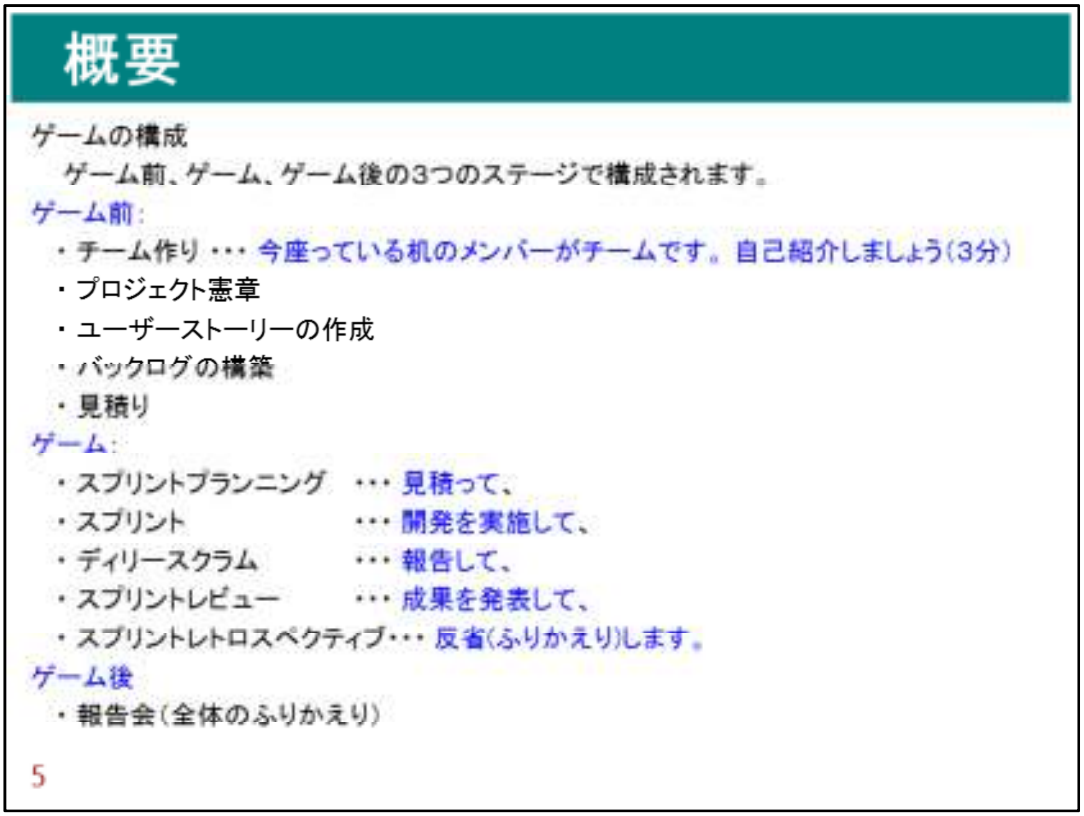
前半のスクラム入門で文房具は確認済みです。
過不足がないか再確認して下さい。

上記以外に、講師には、キッチンタイマーまたは、ストップウォッチが必要となります。
できればカウントダウンタイマーで、時間になると音がなるものがベストです。
また途中で、残り時間を言ってくれる機能があれば更に良いです。

参考として、iPhoneアプリで“こえタイマー”というものがあり、これを使うと円滑に作業を進めることが出来ます。



これから行うスクラムシミュレーションの概要です。
講師がプロダクトオーナーを兼任します。



これから行うスクラムシミュレーションの概要です。
スクラム開発における一連の工程を体験できます。

プロジェクト憲章

1. チーム毎に1つのプロダクトを作ります。チームの皆さんは同じ開発会社の社員です。
2. プロダクトは、ある機能を持つ“街”の開発です。
3. 主な構成要素はLEGOです。他の材料も使えます。
4. 意思決定者は私(講師)です。これは“私の街”です。
5. みなさんの質問に答えたり、これからフィードバックを提供したりするので、私は開発には参加しません。
6. スクラム開発技法を使って、街作りを実施します。

私は、講師(トレーナー)とプロダクトオーナーの2つの役割を受持ちます。
受講者の皆さんは、“開発メンバー”です。

6

レゴブロック（無ければカード）をはじめとして、使えるものをすべて使って、街作りを行います。
この街は、プロダクトオーナーである私（講師）の為の街です。

街の機能

開発する街の名前は、「シリコンビレッジ」です。

以下の建物を街の機能とします。

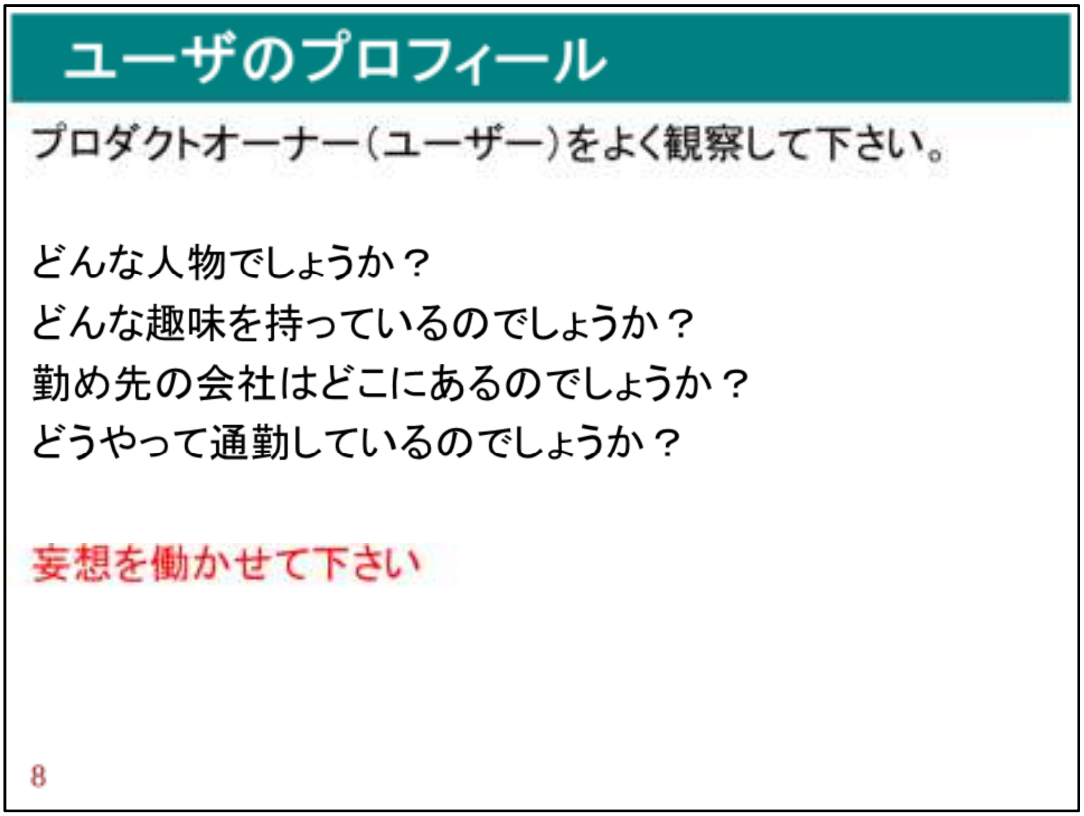
1. 1階建ての建物（1つ）
2. 2階建の建物（自宅）
3. お店1（花屋）
4. お店2（本屋）
5. 学校
6. 教会
7. 病院
8. 幼稚園
9. バス停
10. 交差点（絵でも可）
11. 公園（絵でも可）
12. 川（絵でも可）
13. 橋

地図上に絵を描いても良い機能もあり、その上にLEGOを置く。

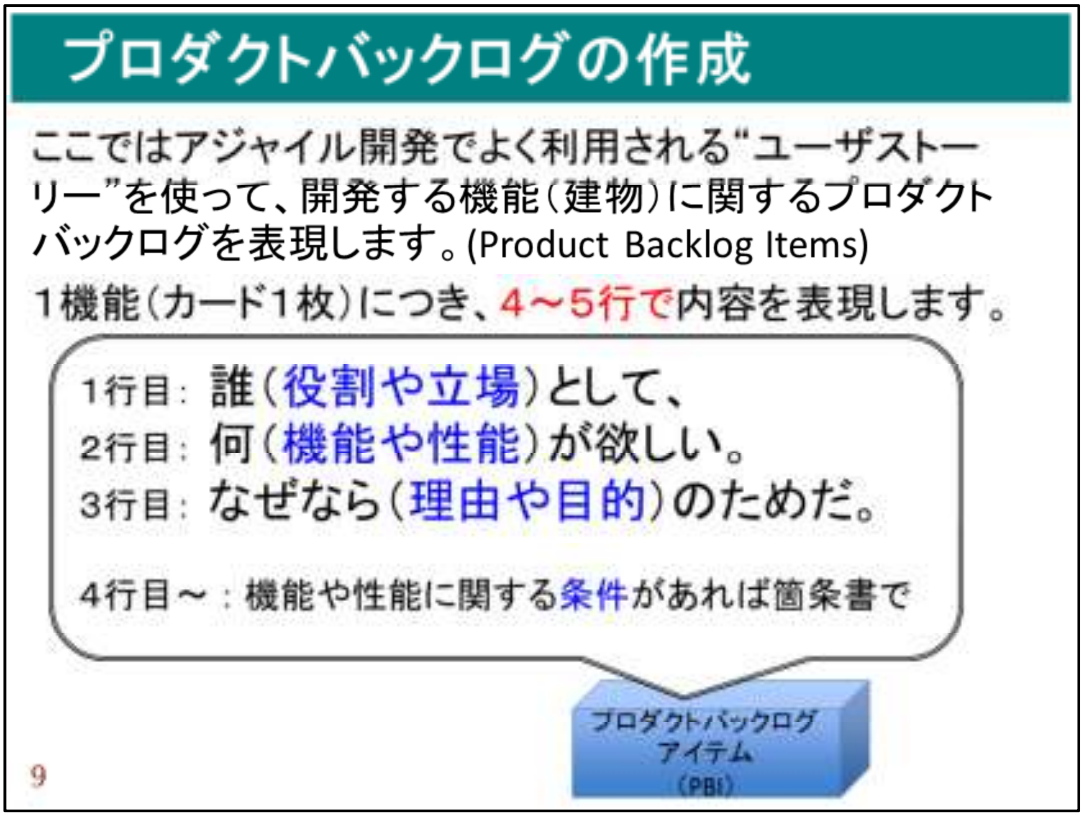
7

私の街は、シリコンビレッジと呼ばれています。
街には、上記の13のモノがあります。
ほとんどは、レゴブロック（または代用品）で作成しますが、模造紙に直接絵で描いても良いものもあります。
交差点、公園、川は絵でも構いませんが、時間が許せば是非レゴブロックで作成して下さい。

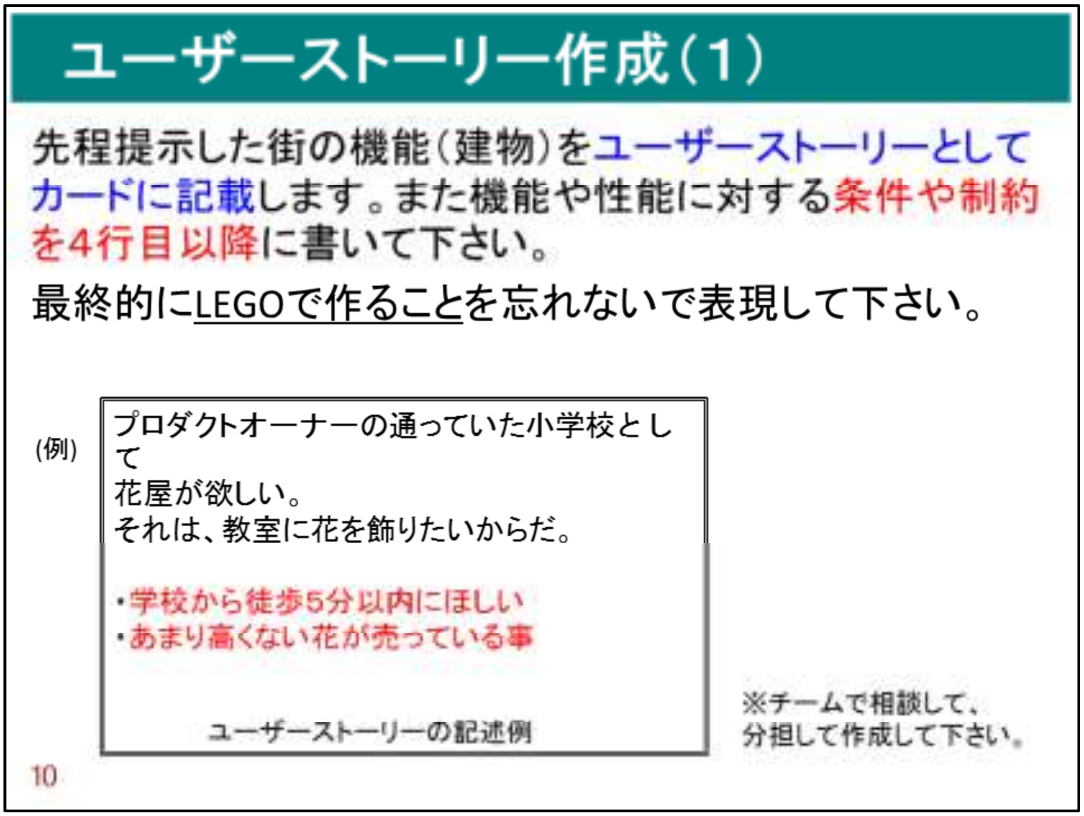
また、みなさんの机の上に貼ってある模造紙は、街全体の土地を表しています。この模造紙の上でバランス良く街作りを行って下さい。



更に追加情報です。
プロダクトオーナーはどんな人物でしょうか？
家族はいるのでしょうか？
好きな食べものは何でしょうか？[外食はするのでしょうか)
どんな会社に勤めているのでしょうか？



これからプロダクトバックログの元となるユーザーストーリーを起こしていきます。
情報カード1枚に付き、1つのユーザーストーリーを書いていきます。本来、プロダクトオーナーの仕事ですが、今回は全員で手分けして書きます。
情報カード上には、4～5行でユーザーストーリーを表現します。



これから手分けして、皆さんでユーザーストーリーを情報カードに書いていきます。
街には13の機能があるので、ユーザーストーリーは13枚作成する事になります。
まずは13個の中から、1人1つの建物を選んで下さい。チーム内で重複しないように皆さんで相談して下さい。（2分間）
レゴブロックで作成するので、建物内部の詳細な記載や細かな装飾などを表現しにくいことを考慮して下さい。

ユーザーストーリーの作成（2）

すべて機能(建物)をユーザーストーリーとして情報カードに記述して下さい。チームで分担して、効率良くすべてのユーザーストーリーを完成して下さい。

情報カードが全て完成したら、全員でユーザーストーリー全体を見てどの機能が難しく、どの機能が簡単かを考えて下さい。

次の見積作業の準備となります。

11

残った建物を分担して、ユーザーストーリーとして起こして下さい。（5分間）
誰がどの建物を書いているか分かるようにチーム内で管理して、重複しない様に気をつけながら13枚のユーザーストーリーを完成させて下さい。

ユーザーストーリーの見積（4）

- スクラムにおける見積は相対値にします
  - ・人月やStepsなどでは考えません
- プランニングポーカーを使う

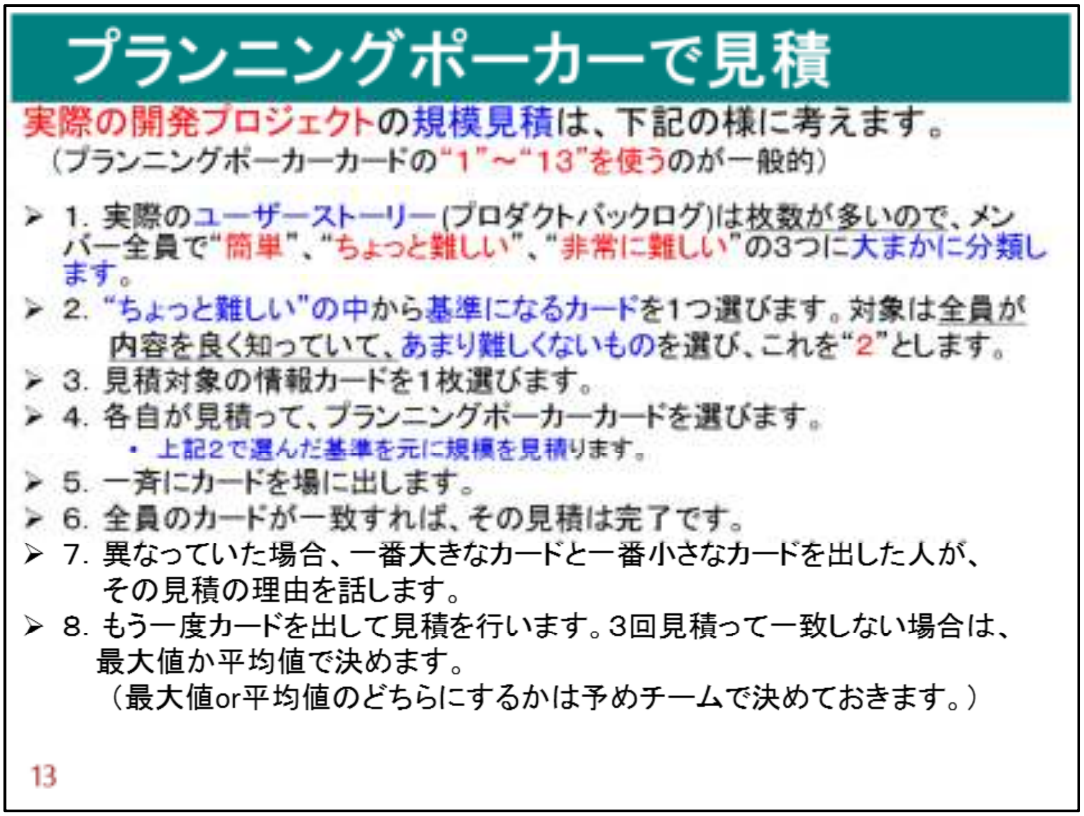
※プランニングポーカーカード
＝フィボナッチ数を記したカード
※フィボナッチ数とは0、1から始まり前の2つの数字を足した数値を表した数列
(0, 1, 2, 3, 5, 8, 13, 21 …)

※市販の名刺シートなどを使えば、自作も可能

【参考】Plus Programming.netコミュニティで公開されているオリジナルプランニングポーカーカード
URL: http://www.survivalplus.net/ja/archives/137

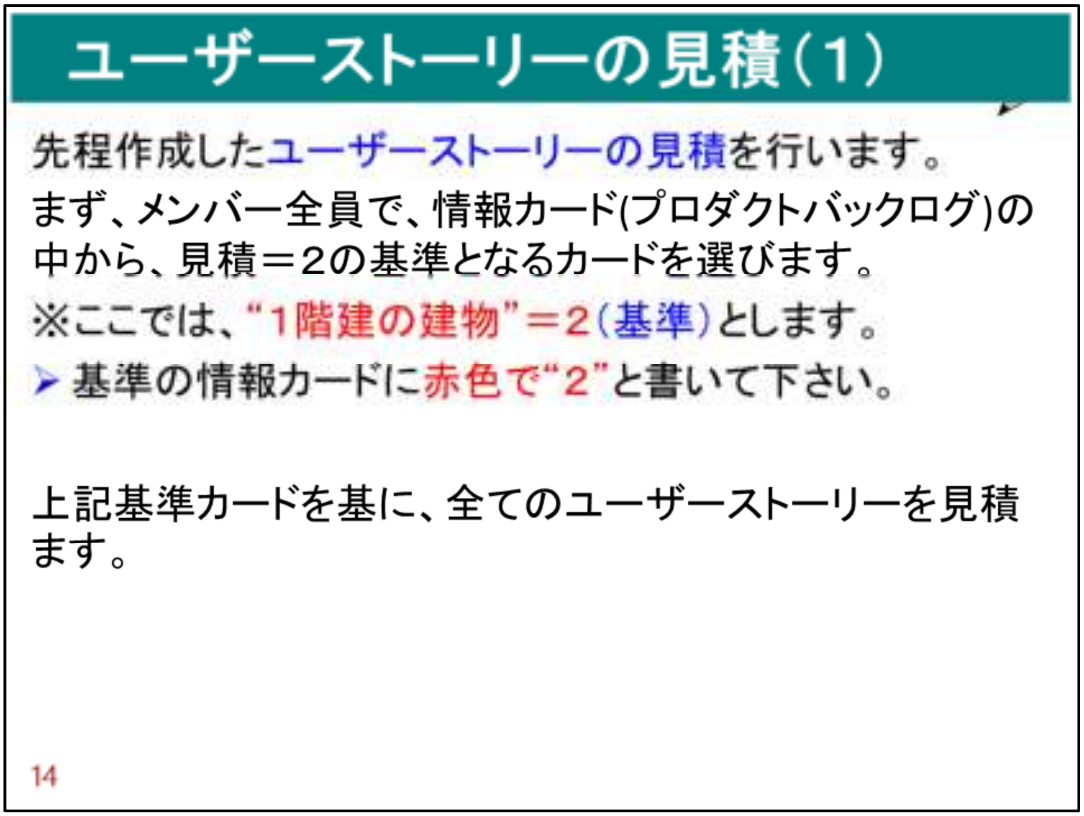
12

このプランニングポーカーカードは、予め人数分＋１セット用意して下さい。
市販の名刺印刷用シートを使えば、簡単に作成する事が出来ます。

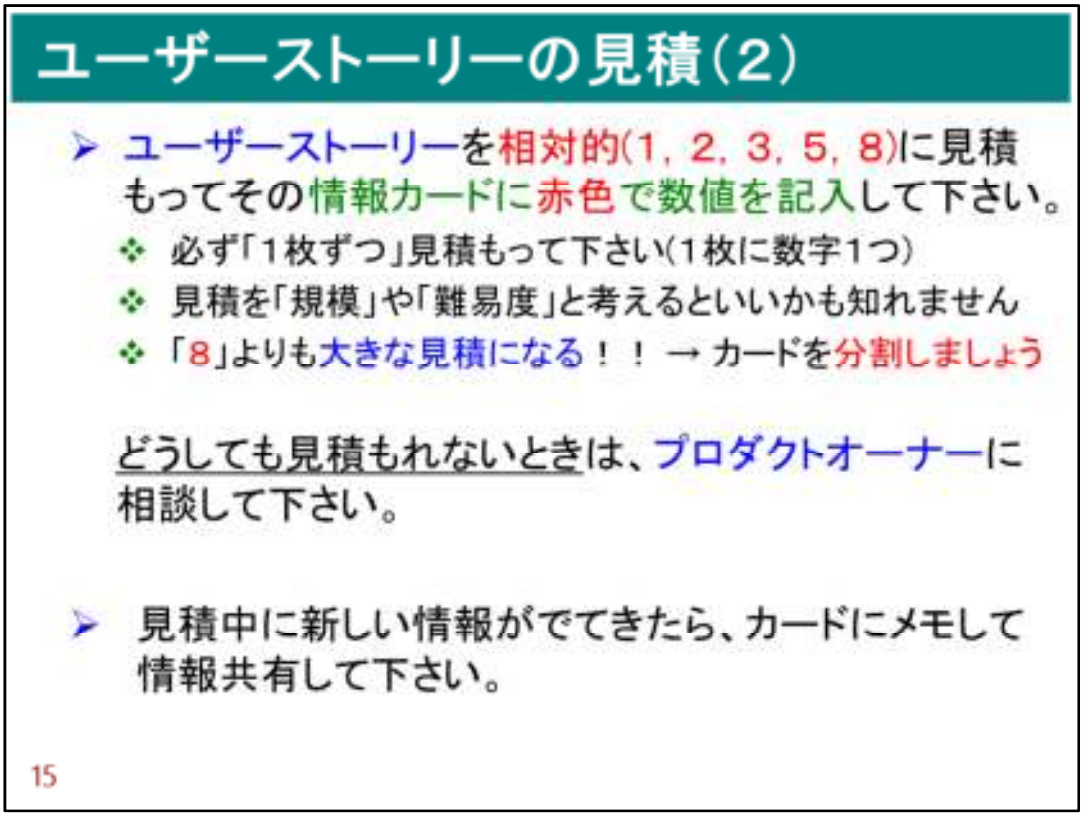


実際の開発プロジェクトでもほぼ同じようにプランニングポーカーを使用した見積を実施します。
但し、1～13のカードを使うことが一般的です。

またユーザーストーリーの数が大変多いので、まずはざっくり3グループに分けて、真ん中のグループから見積を開始した方が効率が良くなります。
また基準のユーザーストーリーを"2"に該当しそうなストーリーから選びます。

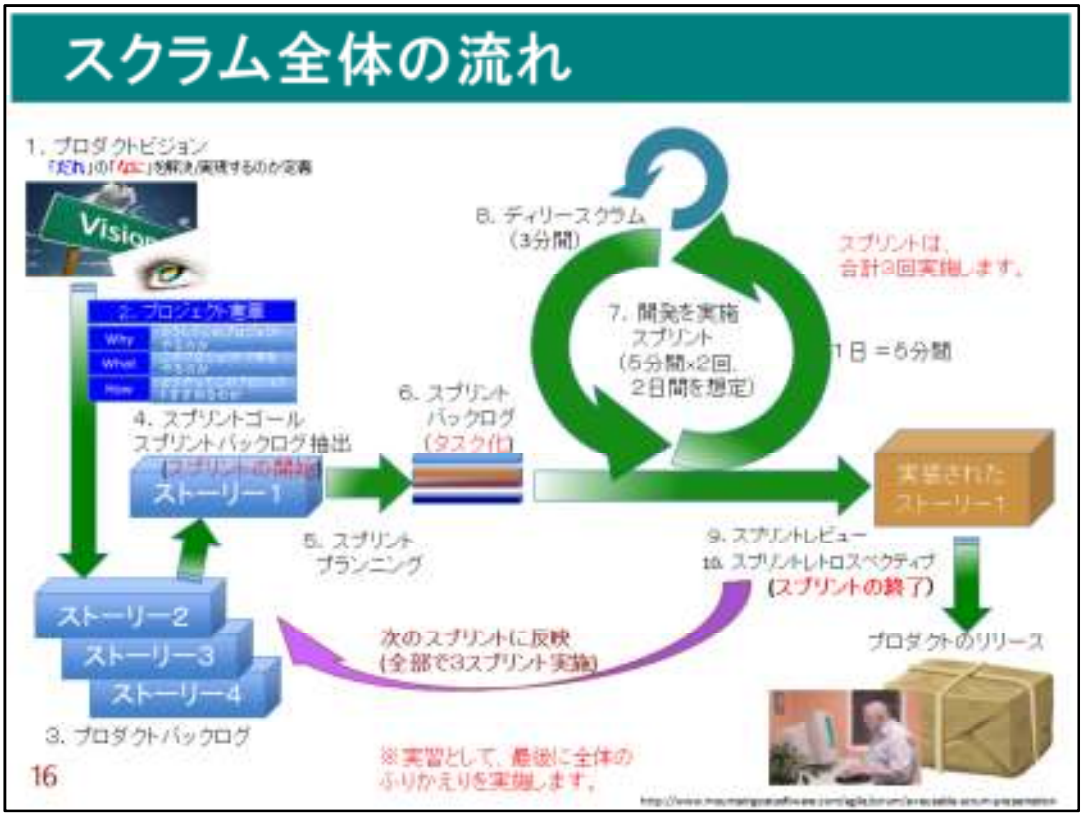


ここからは、先程作成した１３枚のユーザーストーリーを実際に見積もっていきます。



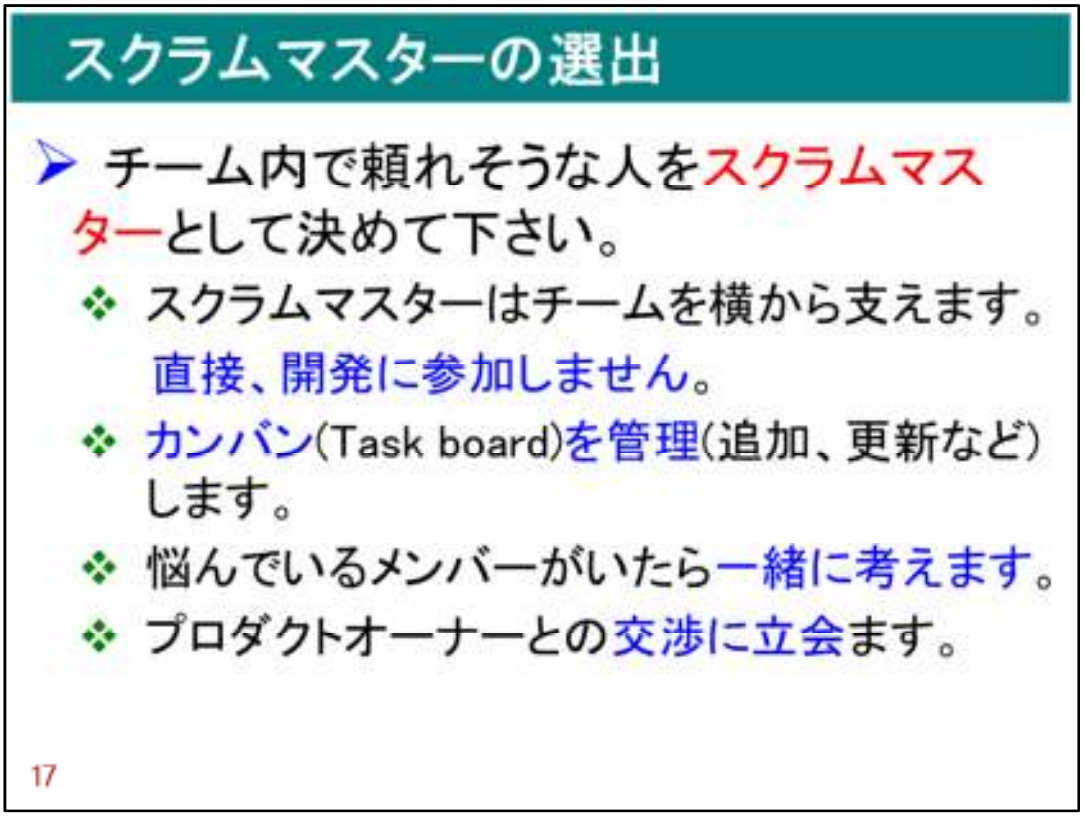
それでは、基準を参考に残り12枚のユーザーストーリーを見積もって下さい。（10分間）
見積が終わったら、ユーザーストーリーの右上に赤色で、見積った数値を書いて下さい。
全ての数値を記入したら、その合計値も求めておいて下さい。

時間がなくなった場合は、多数決で見積を行って素早く決定します。

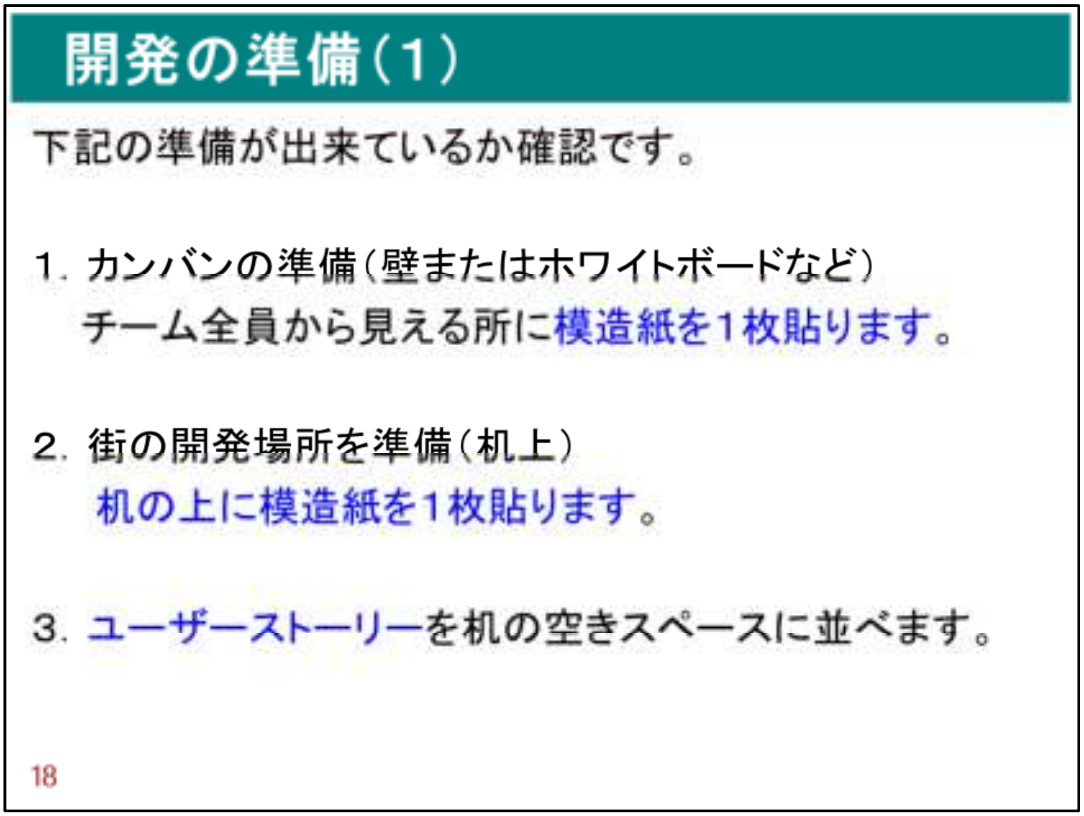


上記№1～10の順番にスクラムの作業が行われます。
特に№4～10はスプリント毎に繰り返して実施されます。

プロダクトバックログ → スプリントバックログ抽出 → 作業タスクへの分割 → 作業を実施 → 結果をタスクに反映 → レビュー → リリース
といった流れで作業が進んでいきます。

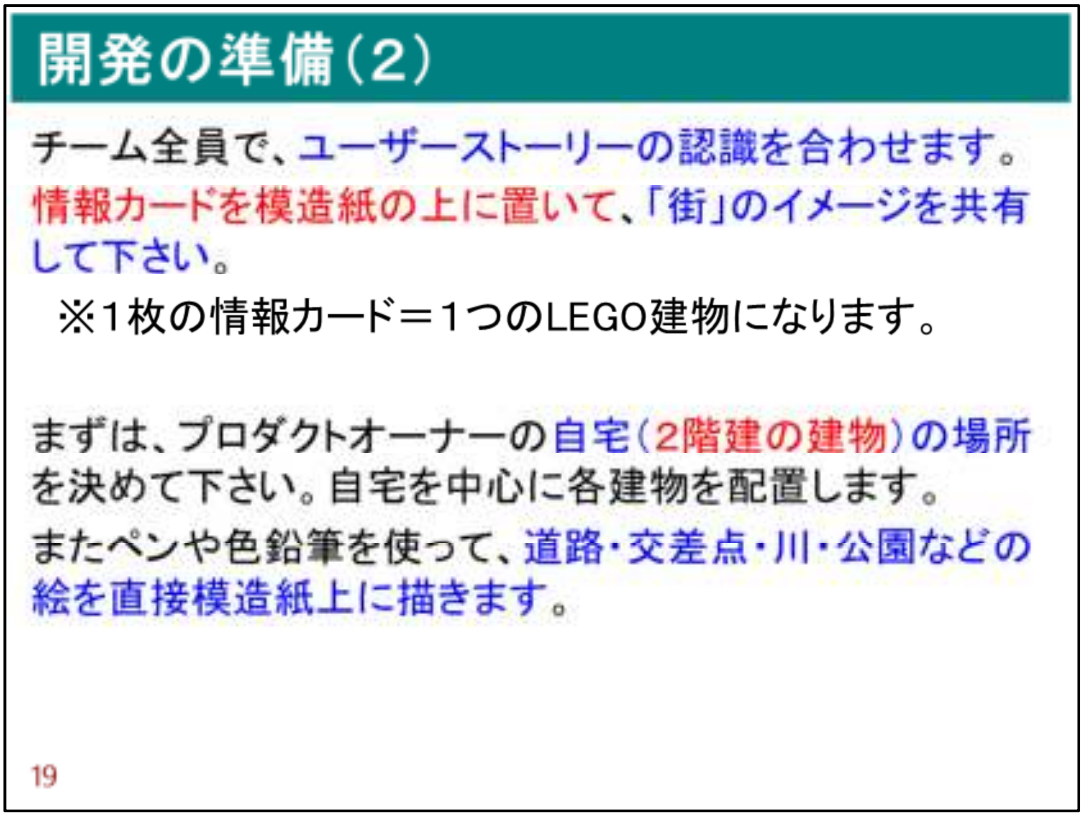


スクラムマスターを各グループで1名決めて下さい。（人数が3名以下の時は選出しない）
スクラムマスタは、街作りに直接参加しません。但し、メンバーの作業を助けることは出来ます。
メンバーが作業に行き詰まっていたら、一緒に考えて下さい。
また、タスクボードと実際の作業が同期しているかをチェックして、タスクボードを適切に管理する事が主な仕事です。
プロダクトオーナーとメンバーの交渉に立ち会って、調停する役割を担います。

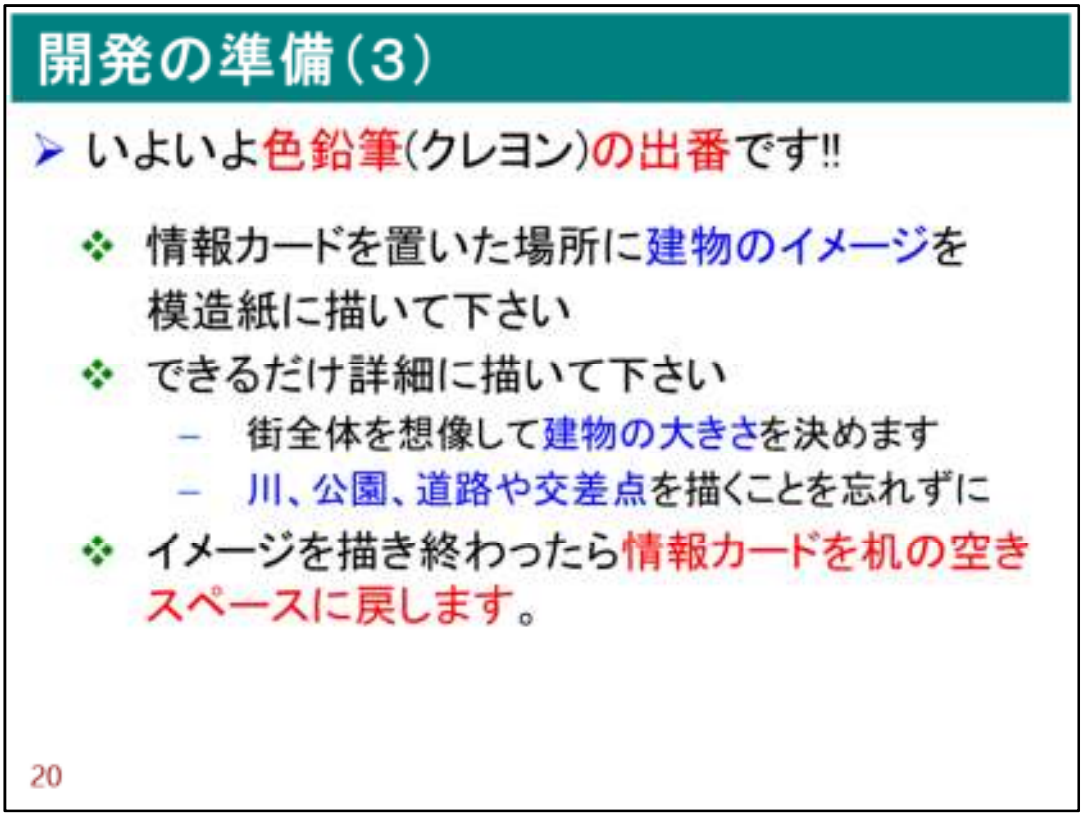


既に準備されているので、再確認します。ずれたりしていたら貼り直して下さい。
模造紙が動かないようにしっかりと止めて下さい。

ユーザーストーリーを机の空いたスペースにすべて移動します。



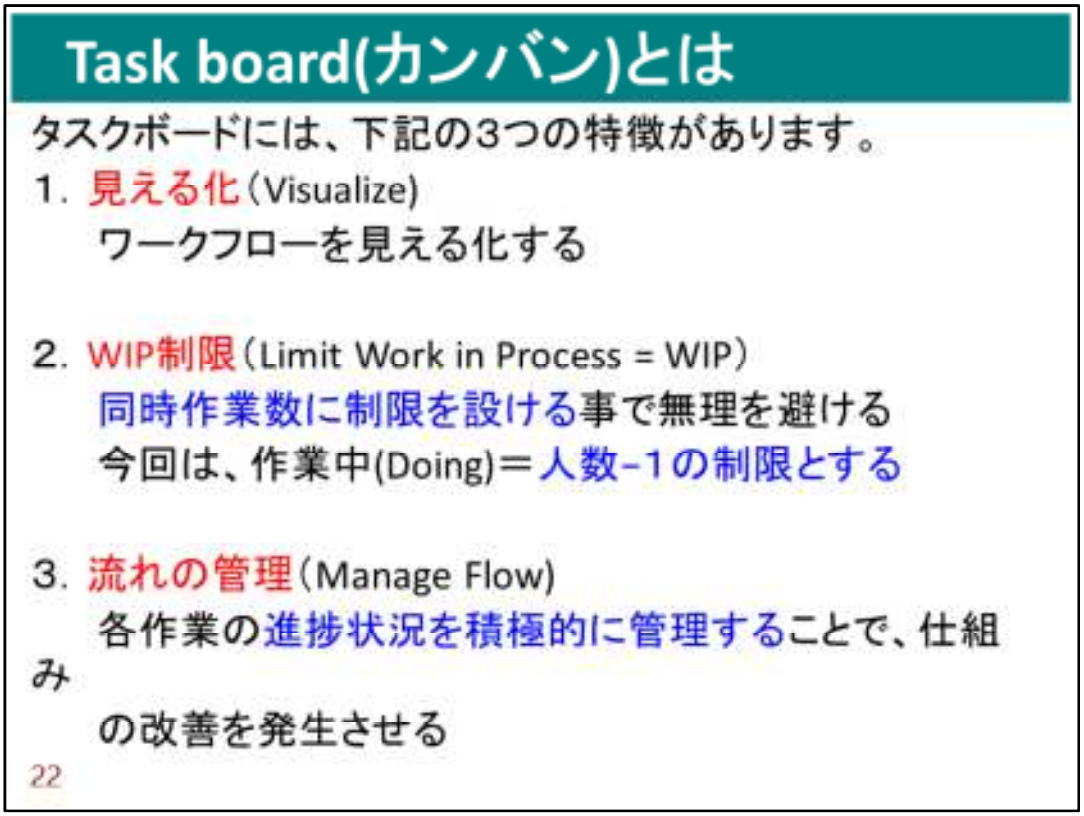
この作業では、情報カード（ユーザーストーリー）を実際の街の原案として、模造紙上に配置して下さい。チーム全員で相談しながら配置してください。（5分間）
またこの時に建物と建物を移動する為に道路や交差点、川などもペンや色鉛筆を使って描いて下さい。道路によって建物がつながり、行き来できる事を確認しながら作業を進めます。
プロダクトオーナーの自宅を最初にどこに配置するかを決めると作業がし易いです。



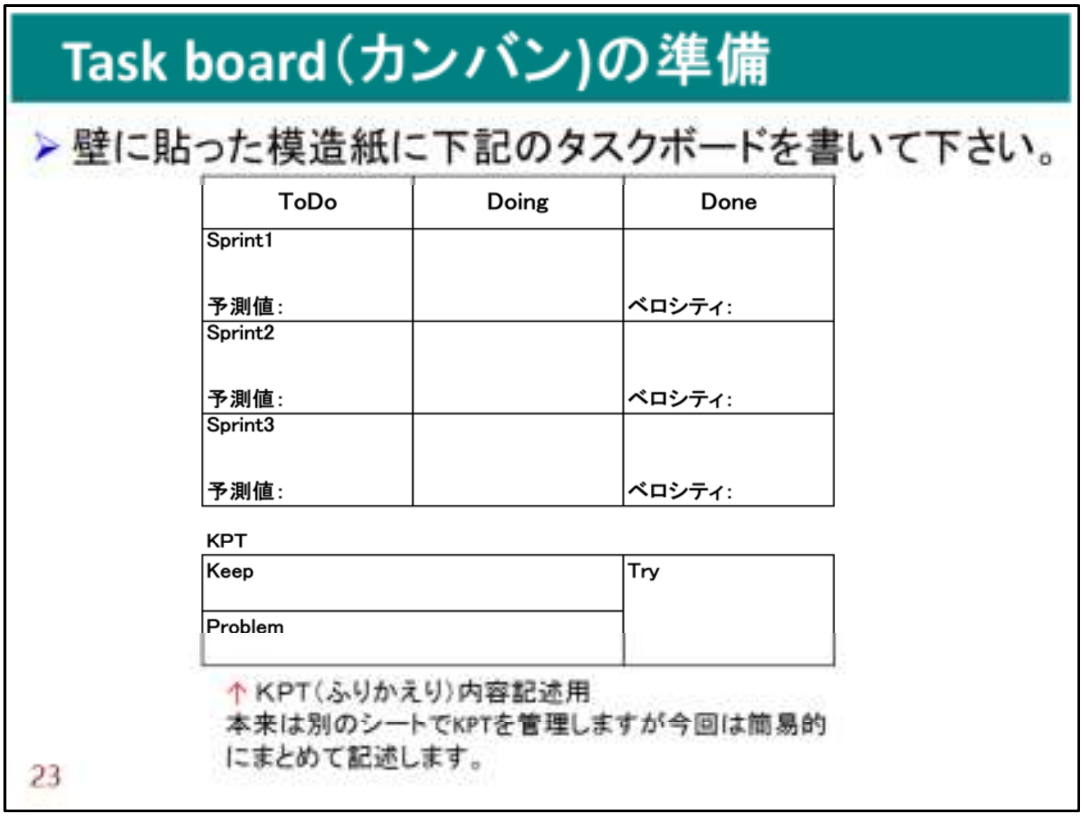
配置したユーザーストーリーの場所に、建物の絵を書きます。（8分間）
どこに何を作るのかを分かる必要があるので、出来るだけ何の建物か分かるように描いて下さい。
その場所にレゴブロックで建物を作って置きます。

絵を書き終わったらユーザーストーリーのカードは、模造紙から外して机の空きスペースに戻して下さい。

付箋紙と模造紙を使ったタスクボード（カンバン）のサンプルです。
ToDo、Doing、Doneが見えます。

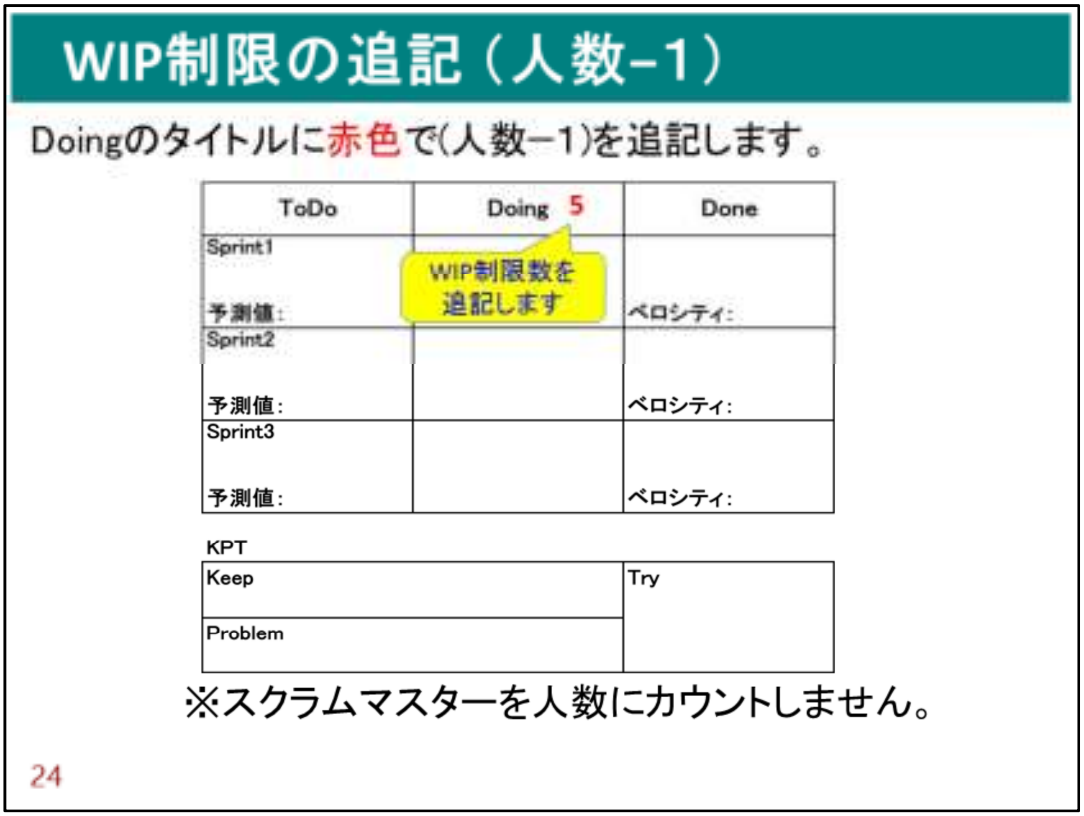


ここではタスクボードの特徴を理解します。基本的にチームの各メンバーが何をどのような状態で作業しているかを一覧できます。
また、WIPと言う作業数の制限もタスクボード上で管理します。これは1人の作業者が、複数の作業を同時に抱えてしまわないようにする為の制約です。スクラムマスター分の除く為、WIP＝チームの人数-1とします。※スクラムマスターを立てない場合、WIP＝チームの人数とします。
タスクとしては、このプロジェクトに関するものも、関係しないものもすべて表現するのが良い。
そうすることで、プロジェクト以外の作業タスクがどれ位有り、本当にいま必要なのかも判断し、必要なければスクラムマスターを通して外部と交渉してもらう等の手段を講じる事が出来る。



上記の様に壁もしくはホワイトボードに貼ってある模造紙上にこのような表を作成して下さい。(3分間)
メンバー全員で手分けして作成して下さい。
予測値とは、実施できる見込値であり、ベロシティは実施できた実績値を指します。

最下行は、ふりかえり(レトロスペクティブ)用KPT(Keep、Problem、Try)エリアです。



前述のWIPをDoing欄に赤色で記入します。
この値までしか、付箋紙をDoingエリアに同時に貼れない制限となります。

KPTの使い方

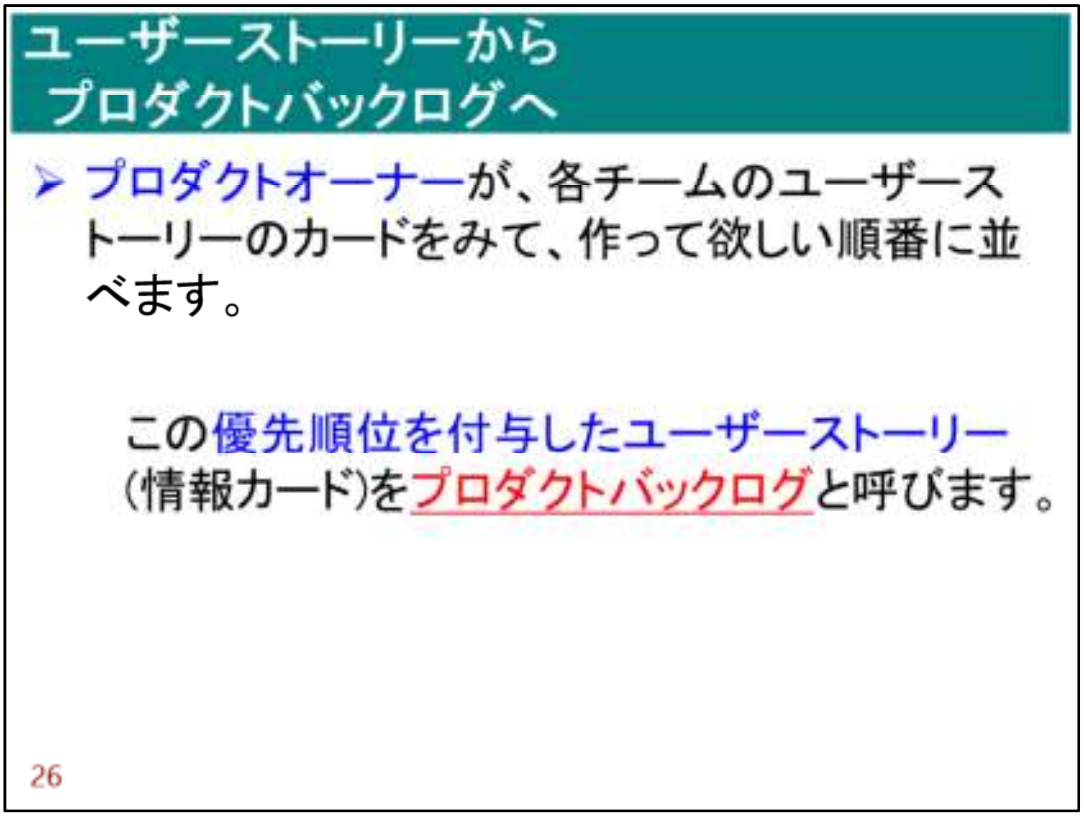
- 開発作業をはじめる前に、疑問点や不安点を話し合って付箋紙でKPTの「Problem」枠内に貼ってください。

（例）

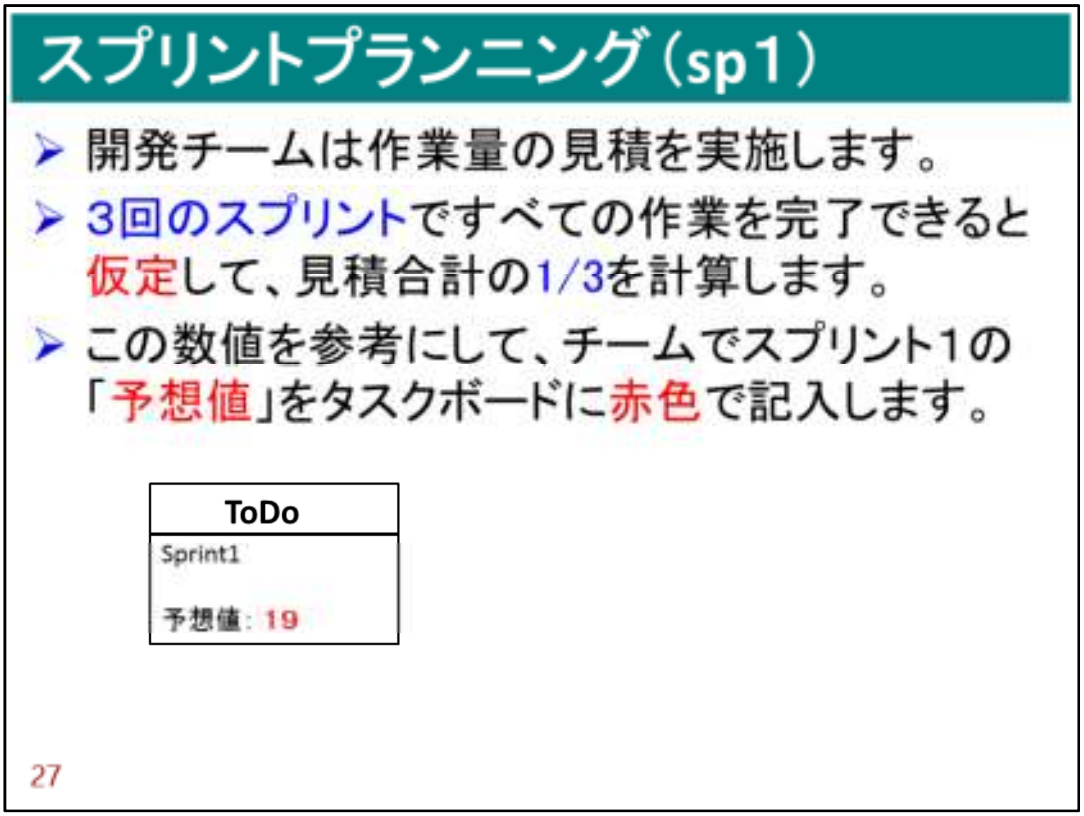
- LEGOでこの建物は表現できるか？
- 白色のブロックは足りるのか？
- 病院作成の見積は甘く無いか？　　など

25

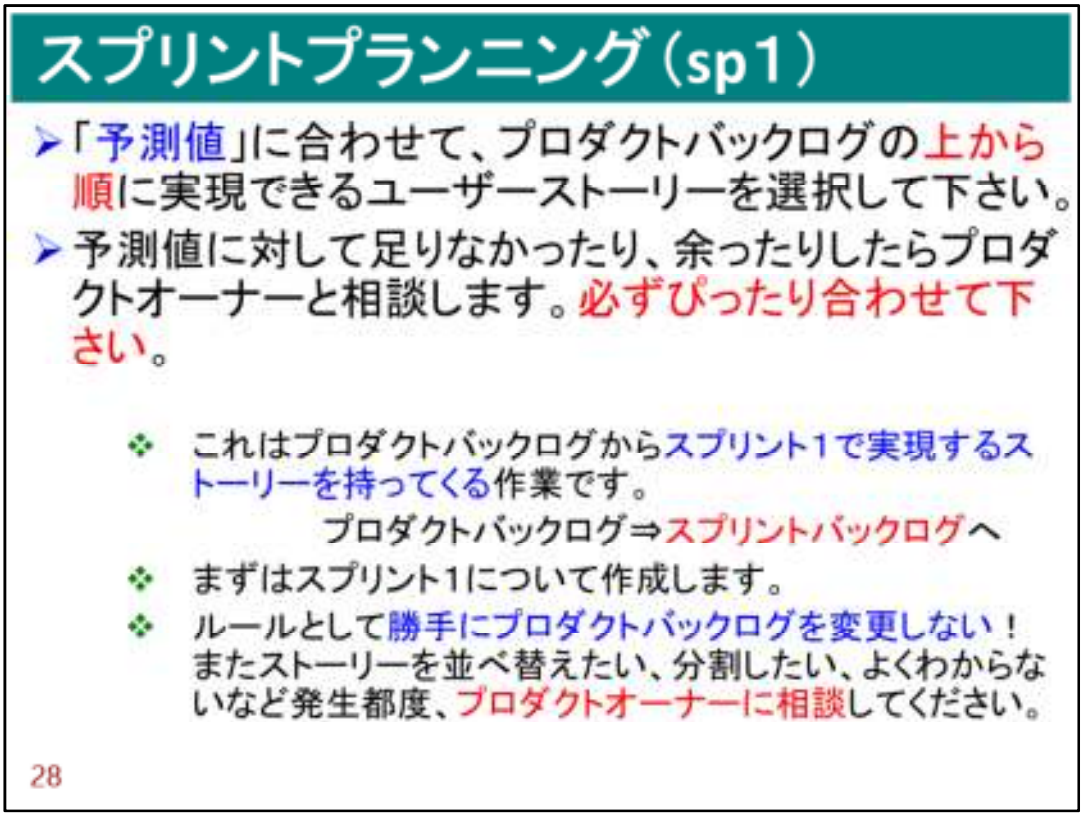
作業開始前に想定される疑問点や問題点などあれば、付箋紙に記入して、Problem欄に貼っておきます。（2分間）



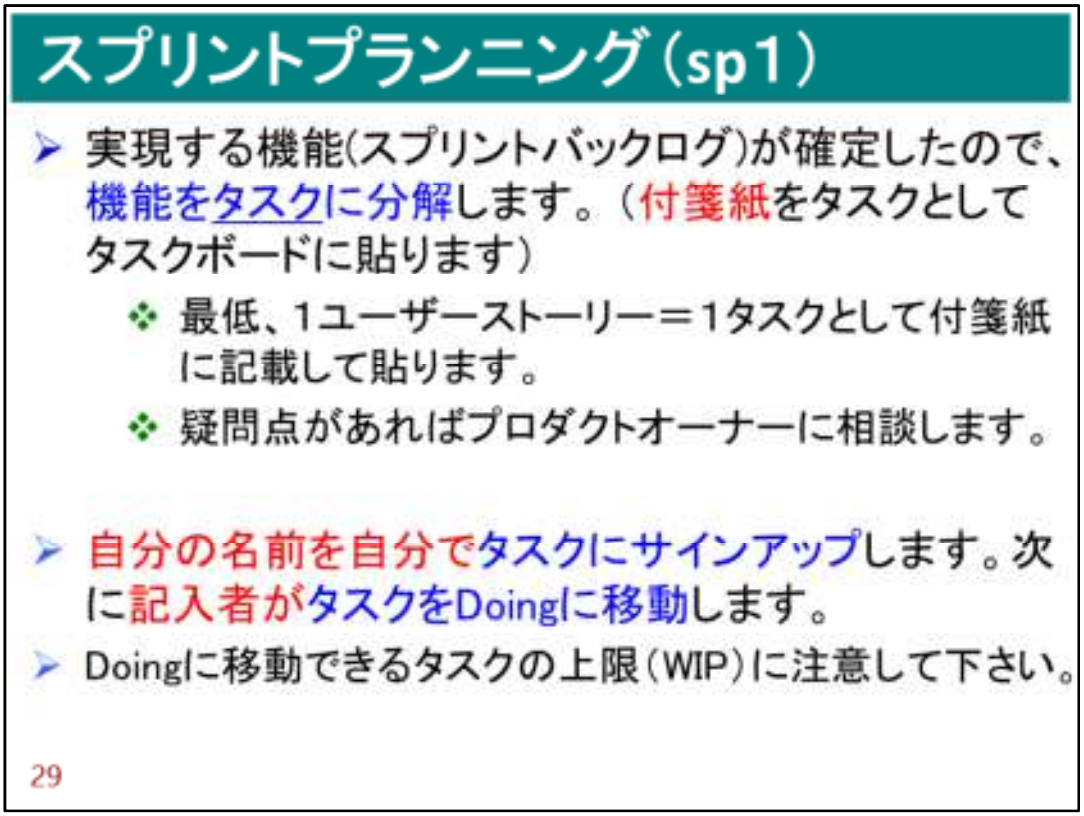
プロダクトオーナー(講師)が、テーブルに移動して、ユーザーストーリーを作成して欲しい順に並べます。
これがユーザーストーリーの優先順位になり、優先順位付きのユーザーストーリーをプロダクトバックログと呼びます。



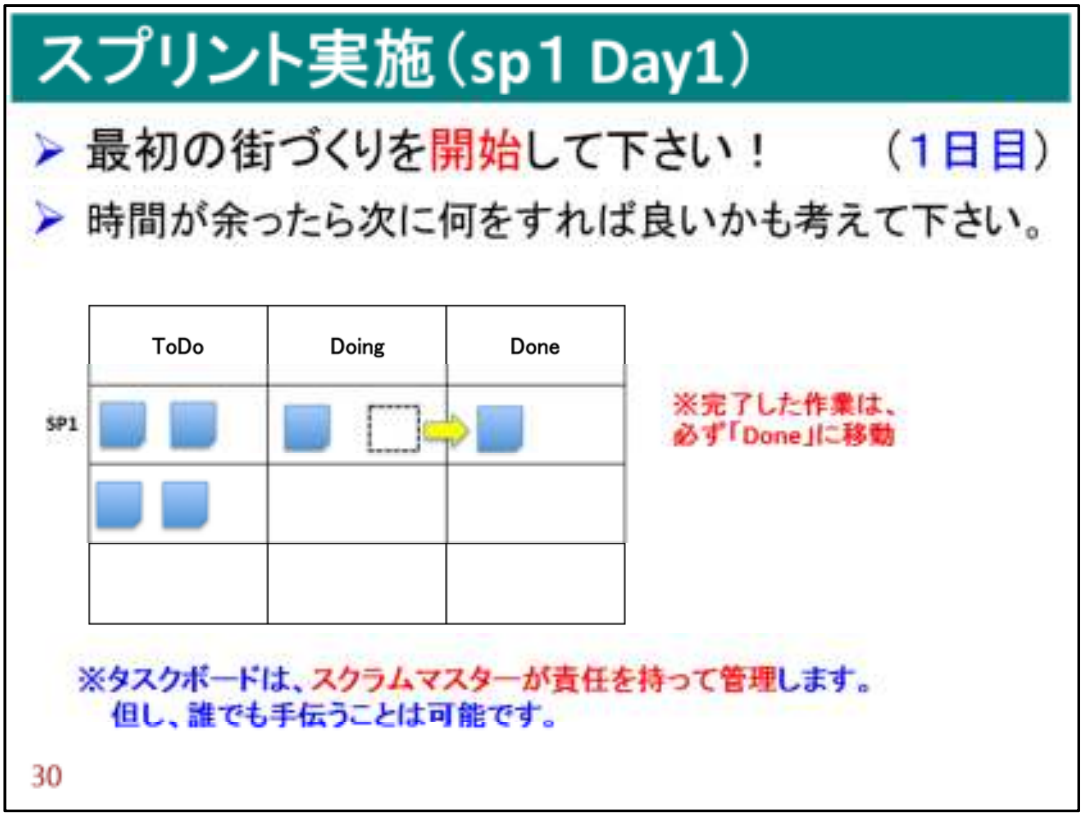
まずは、全ユーザーストーリーの見積値の合計を求めます。（2分間）
次に3スプリントで完了すると想定しているので、見積合計値を3で割ります。この値をスプリント1の予想値とし、赤色で記入します。



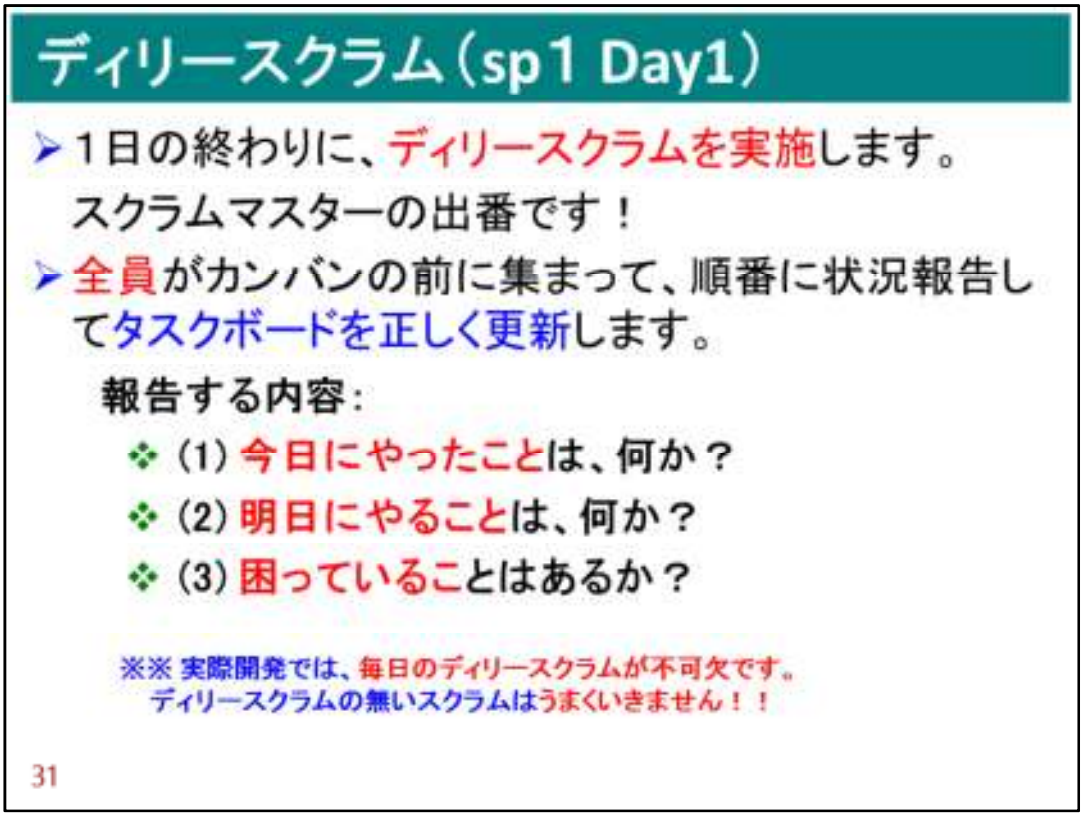
スプリント1では、予想値まで作業が出来ると想定しているので、予想値と合致するようにユーザーストーリーを上から順番に選んでいきます。（3分間）
これがスプリント1のスプリントバックログになります。もし予想値にぴったりにならない場合は、プロダクトオーナーに相談して下さい。
プロダクトオーナーは、優先順位を並べ替えるか、ユーザーストーリー分割するかして、予想値ぴったりになる様に調整します。
ユーザーストーリーを変更できるのは、プロダクトオーナーだけです！！



次にタスクボードの前に集まって、スプリント1で行う実際の作業をタスクとして書き出します。（3分間）
白色ブロックを集める、病院の1階部分を作るなど、1人が10分位で出来そうな単位に分解してタスク化します。
わからなければ建物名を直接書いて置きます。
タスク化が終わったら、自分で行いたいタスクに、自分の名前を書きます。（サインアップ）
名前を書いたら作業開始なので、その付箋紙をDoingに移動します。

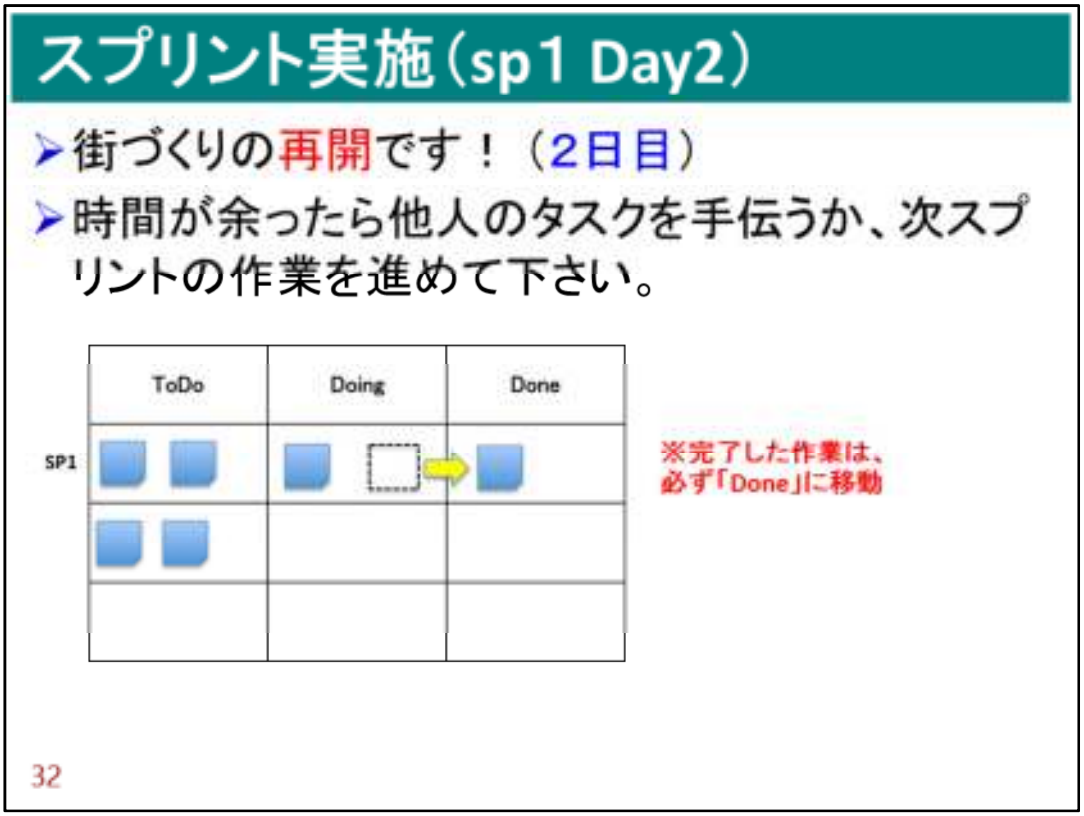


これから5分間で街作りを開始します。（5分間）
タスクに従って作業を開始して下さい。作業が完了したらタスクの付箋紙をDoneに移動して、次のタスクをサインアップし、作業を継続して下さい。（基本は自分で付箋紙を移動します）
スクラムマスターは、タスクボードと作業が一致しているかを観察し、ずれている場合、訂正して下さい。
またスクラムマスターはチームメンバーの作業を手伝うことも出来ます。

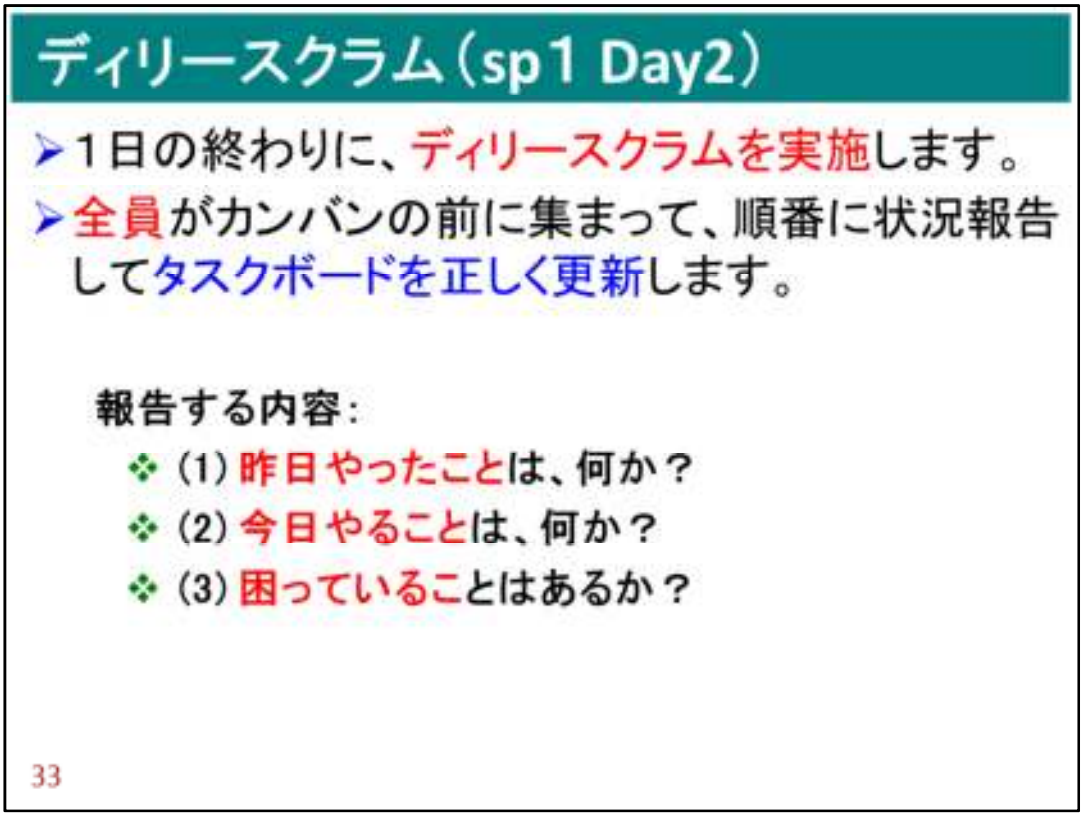


スプリント1の1日目が終了しました。（3分間）
全員、タスクボードの前に集まって、ディリースクラムを行います。
順番に上記3つの点について報告し、必要に応じてタスクボードも更新して下さい。
スクラムマスターはディリースクラムの進行役を務めて下さい。

スクラム開発において、ディリースクラムはとても重要です。

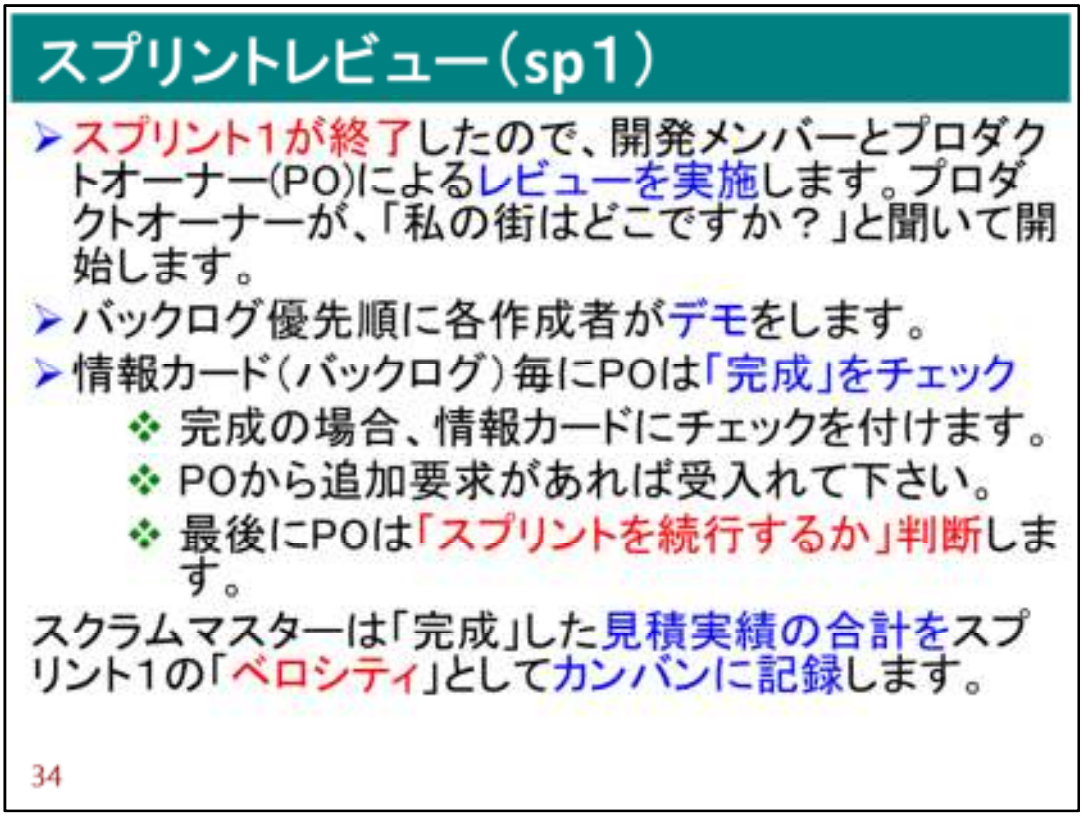


スプリント1の2日目が始まります。（5分間）
1日目と同じように作業を開始して下さい。
この5分間が終わるとスプリント1の開発が終了します。
街の模造紙の上に乗っているもののみレビュー対象として評価します。



スプリント1の2日目が終了しました。（3分間）
全員、タスクボードの前に集まって、ディリースクラムを行います。
順番に上記3つの点について報告し、必要に応じてタスクボードも更新して下さい。

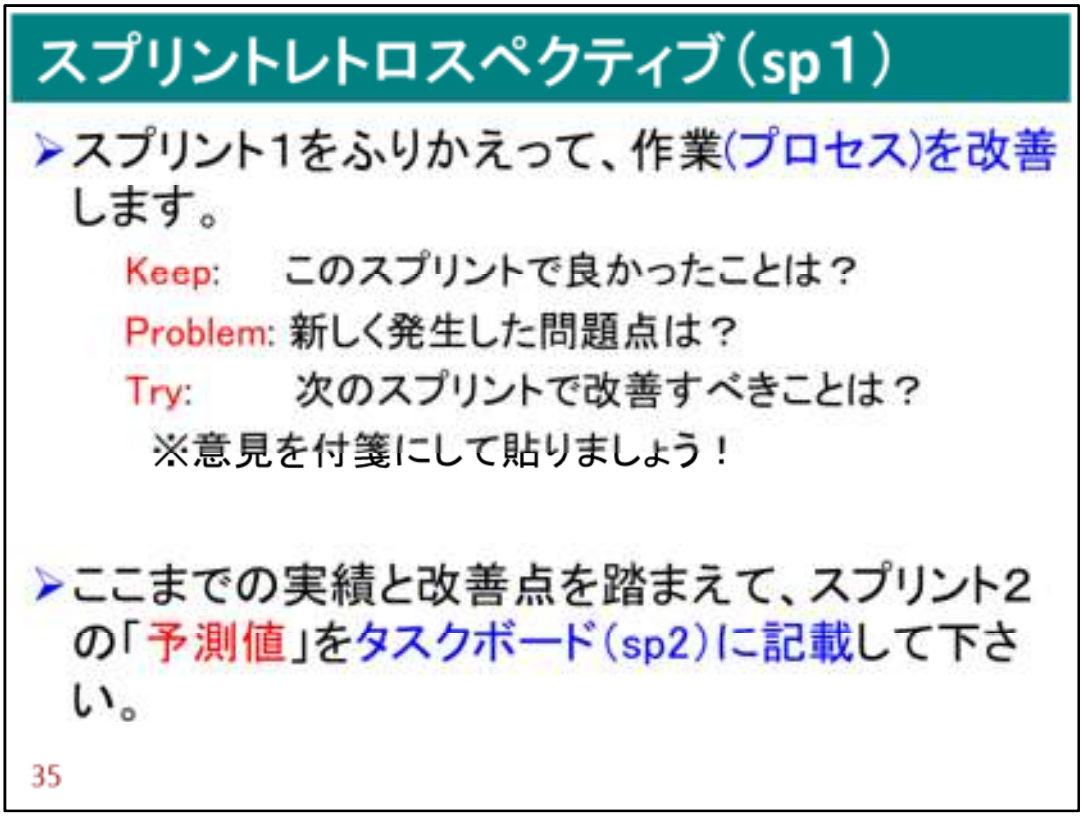
スクラム開発において、ディリースクラムはとても重要です。



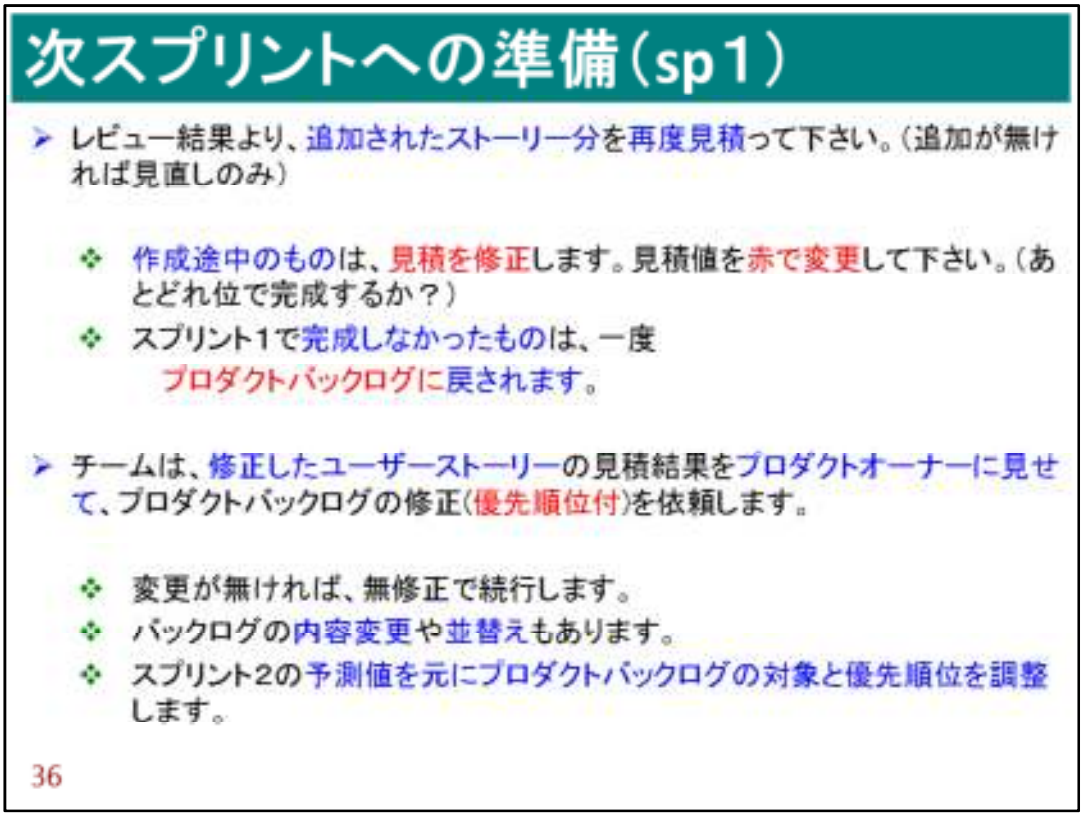
全員（全てのチーム）が街の前に集まってスプリントレビューを開始します。（5分間 x チーム数分）
レビューは、建物開発者がプロダクトオーナーに対して、建物を説明（デモンストレーションします。
プロダクトオーナーはその説明を聞いて、出来具合を判断します。
完成している場合：
　その建物を受入れるか／やりなおしかを判断します。
　　→ 受け入れた場合、ユーザーストーリーに「完」のマークをつける。
　　→ やり直しの場合、その理由を確認してユーザーストーリーに追記する。

まだ完成していない場合：
　開発を続行するか／中断するかを判断します。
　　追加要求をつけた場合 → ユーザーストーリーに追加要求を追記して、継続。終わった部分の見積値を算出して、残り数を赤で書き入れる
　　→中断する場合：その建物の中断理由を聞いて、ユーザーストーリーに追記し、この建物の作業を凍結する。

完了した作業の合計実績を計算して、この値をスプリント1のベロシティとして、赤色で記載する。

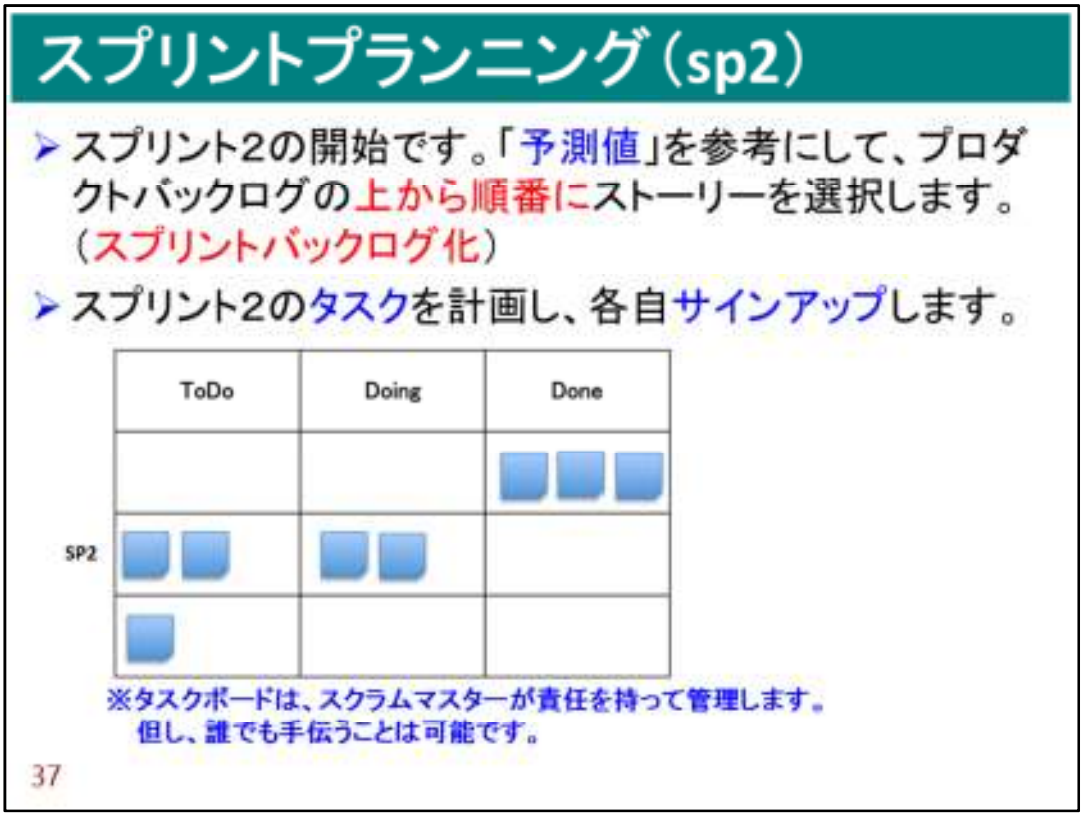


レトロスペクティブの実施。（3分間）
スプリント1の作業をチームで振り返って、KeepとProblem、Tryに当たる内容を付箋紙に書き出し、タスクボードに貼る。
メンバー全員で、スプリント1の予測値とベロシティを参考にして、スプリント2の予測値を決定しタスクボードに赤色で記載する。
スクラムマスターが会議を主催します。

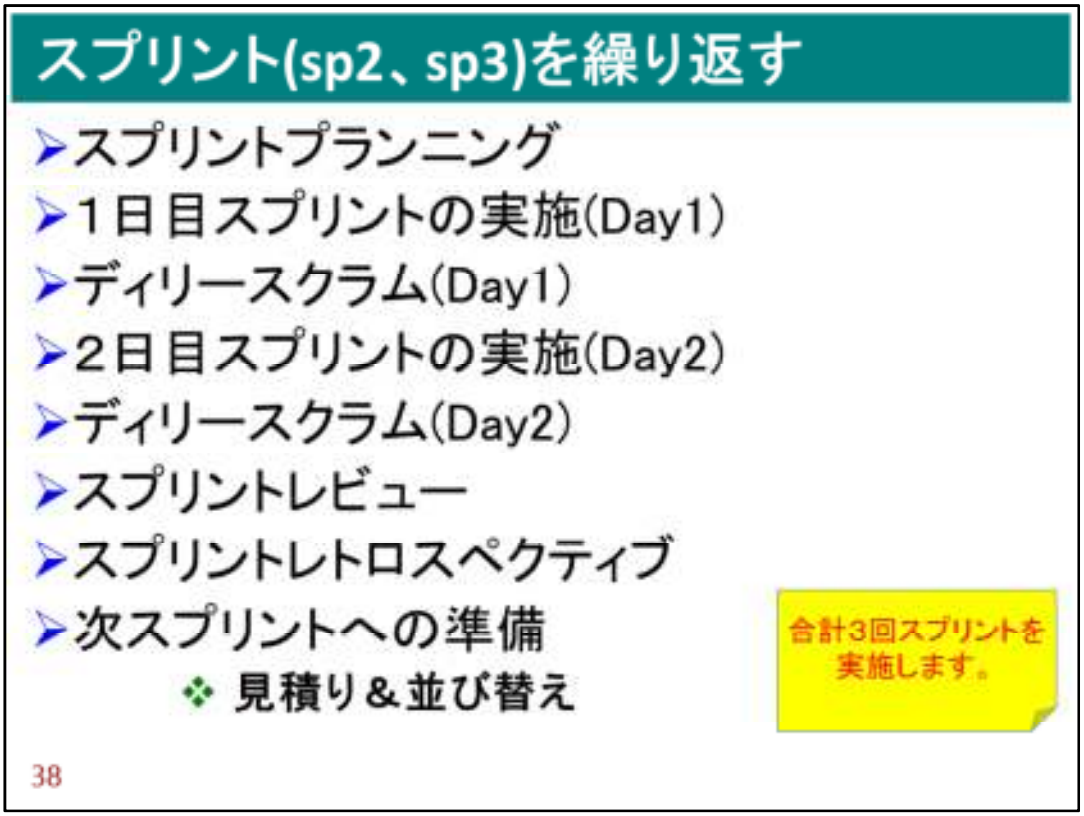


プロダクトオーナーより追加された要望等があれば、ユーザーストーリーの見積を見直して値を変更します。（3分間）
スプリント1が終了したので、未完成のユーザーストーリーは、そのままプロダクトバックログに戻されます。

変更した内容をプロダクトオーナーに伝えて、プロダクトバックログの優先順位に変更が無いかどうかを判断してもらいます。
スプリント2に向けて、スプリント2で開発対象とするユーザーストーリーを抽出し、予測値と合致するかを検討する。

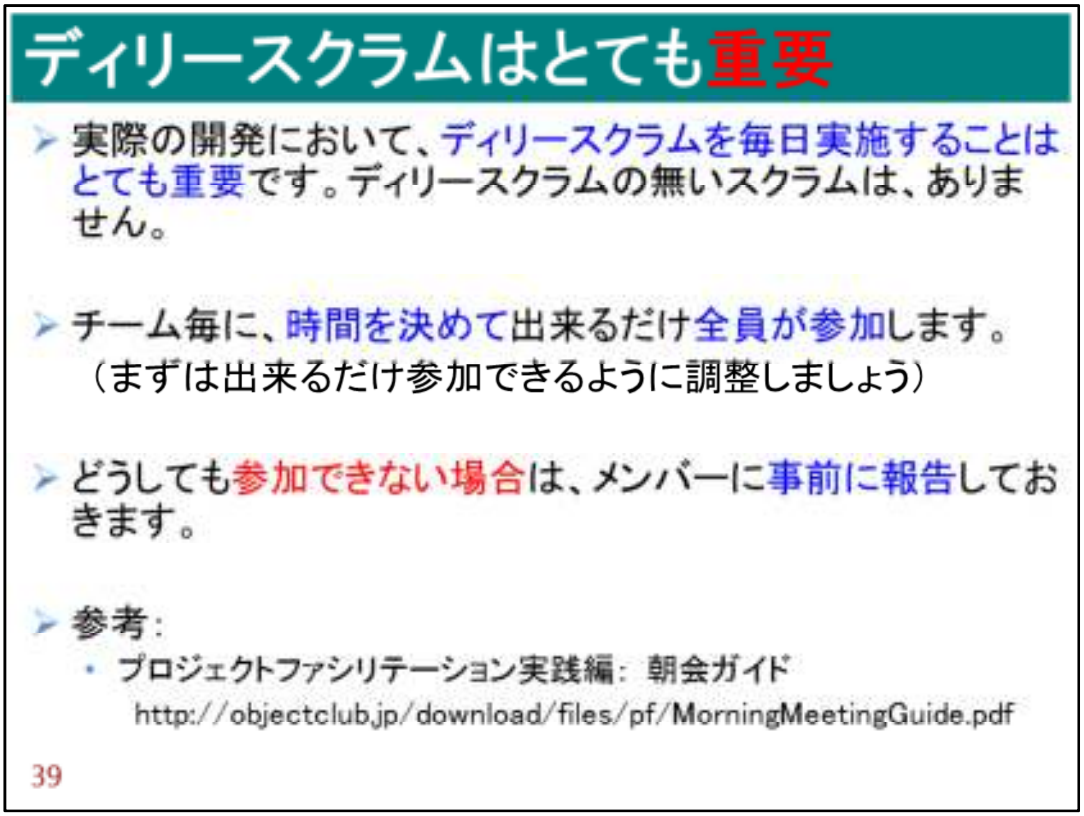


スプリント2が開始され、予測値と合致するようにユーザーストーリーの調整をプロダクトオーナーと行う。（3分間）
次にタスクボードの前に集まって、タスクを作成し、サインアップする。

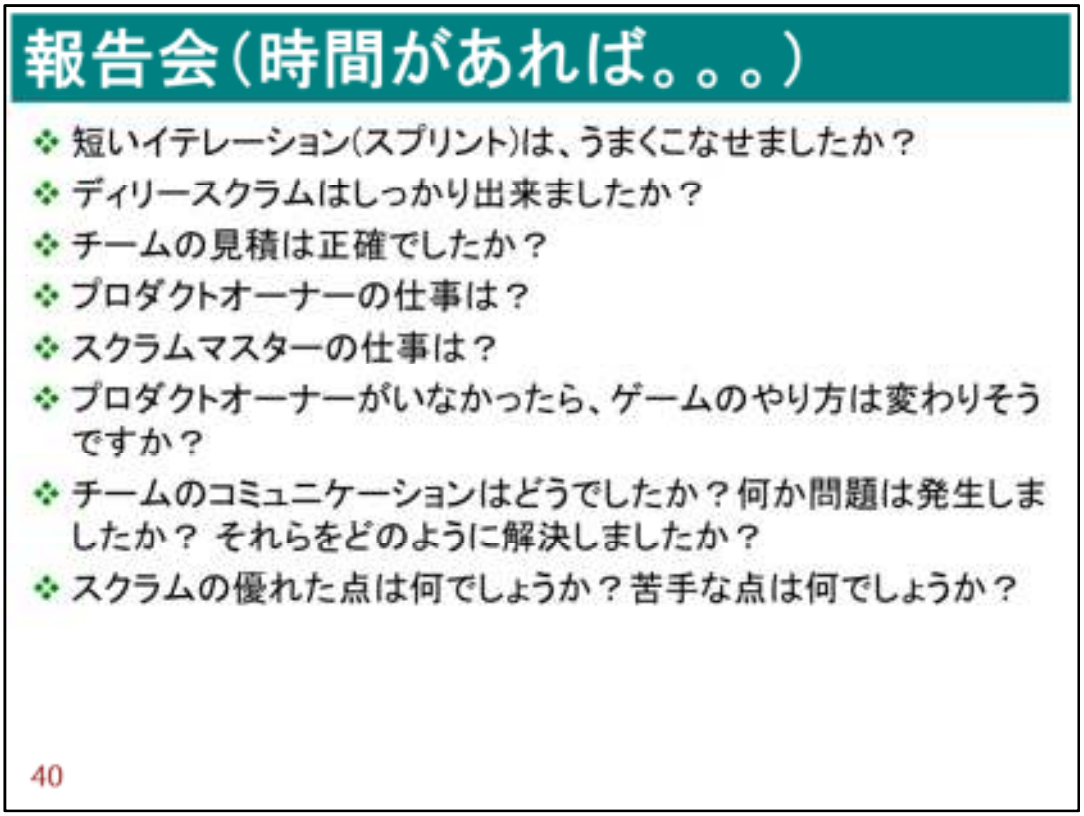


以後は、スプリント1と同様に進めて行く。
全部で3回のスプリントを実施し、すべての開発を終了する。

1. スプリントプランニング（3分間）
2. 1日目スプリント実施（5分間）
3. ディリースクラム(Day1)（3分間）
4. 2日目スプリント実施（5分間）
5. ディリースクラム(Day2)（3分間）
6. スプリントレビュー　（5分間 x チーム数分）
7. レトロスペクティブ　（3分間）
8. 次スプリントへの準備（3分間）



スクラム開発手法では、メンバーの自主性を尊重して進めているので、毎日の進捗を発表しあう、ディリースクラムの重要性が高い。
ディリースクラムを行わないとメンバーがバラバラに作業を実施していることになり、チームとしてのまとまりが無くなり、開発チームが分裂する危機が容易に訪れます。
長くても15分程度のミーティングなので、できるだけ全員で実施して下さい。



ここまでのスクラム入門とスクラムシミュレーションを通して、みなさんが気づいた点を述べて下さい。
上記の各項目は、あくまでも参考です。フリートークで感じたことを表現して下さい。
スクラム開発の簡単だと思われる点や難しいと思われる点について、述べられても構いません。

これでスクラム入門の講座を完了します。

地方型グローバルIT人材の育成 PBLカリキュラム
Redmin（BackLogs）の使用方法

・本プロジェクトにおいてのRedmineの使用方法を説明します。

e-learningでスクラム全体の流れが理解できていることが前提です。

スクラム開発手法の以下の道具がRedmineでどのように使用するかをポイントに説明します。

【3つのロール（役割）】
(1)プロダクトオーナー ・・・ 製品の責任者
(2)スクラムマスター ・・・ スクラムの番人、チームを守る
(3)開発チーム ・・・ プロジェクトにフルタイム参加する開発メンバー

【4つのミーティング】
(1)スプリント計画会議 ・・・スプリントのゴールとスプリントバックログを決定
(2)ディリースクラム ・・・ 毎日15分の報告会（3つの質問）
(3)スプリントレビュー ・・・ 完成したアプリのレビューを実施
(4)ふりかえり ・・・ うまくいった事、いかなかった事などを報告
（Retrospective） （Keep:継続すること, Problem:やめること, Try:やること）

目次

1.Redmineの概要 ……4
2.Redmineの機能概要 ……5
3.プロジェクト体制 ……6
4.担当と役割 ……7
5.プロジェクトにおける役割 ……8
6.プロジェクト情報 ……9
7.グループ情報 ……10
8.ユーザー情報 ……11
9.ユーザー情報　変更方法 ……12
10.Scrumフレームワーク ……13
11.トラッカー（チケットの種類） ……14
12.スクラム全体の流れ ……15
13.役割別の利用ツールと利用機能 ……16

14.スプリントプランニング ……18
14.1 スプリントプランニング（プロダクトバックログ作成） ……19
14.2 スプリントプランニング（スプリントバックログ作成） ……24
14.3 スプリントプランニング（タスク作成） ……32

15. スプリント……40
15.1 スプリント（タスクの更新） ……41
15.2 スプリント（タスク成果物） ……43
15.2 スプリント（デイリースクラム） ……46
15.3 スプリント（作業時間の記録） ……47
15.4 スプリント（スプリント障害タスクの入力） ……50
15.5スプリント（バーンダウンチャート） ……52

目次

16 スプリントレビュー ……53
16.1 ストーリーポイントの見直し ……54
16.2 スプリントレビュー（結果の記録）……56

17 振り返り(KPT) ……57
17.1 振り返り(KPT)（作業時間の記録を参照）……58
17.2 振り返り(KPT)（結果を記録）……59

本プロジェクトではプロジェクト管理ツールとしてRedmineというプロジェクト管理ツールを使用します。
RedmineはWebベースのプロジェクト管理システムです。

本資料は、Surum開発を支援するプラグインである「Backlogs」の使用方法について説明します。
Redmineの構築方法、設定方法は別教材で準備しています。

Redmineの基本的な操作は、以下のサイトでも確認できます。

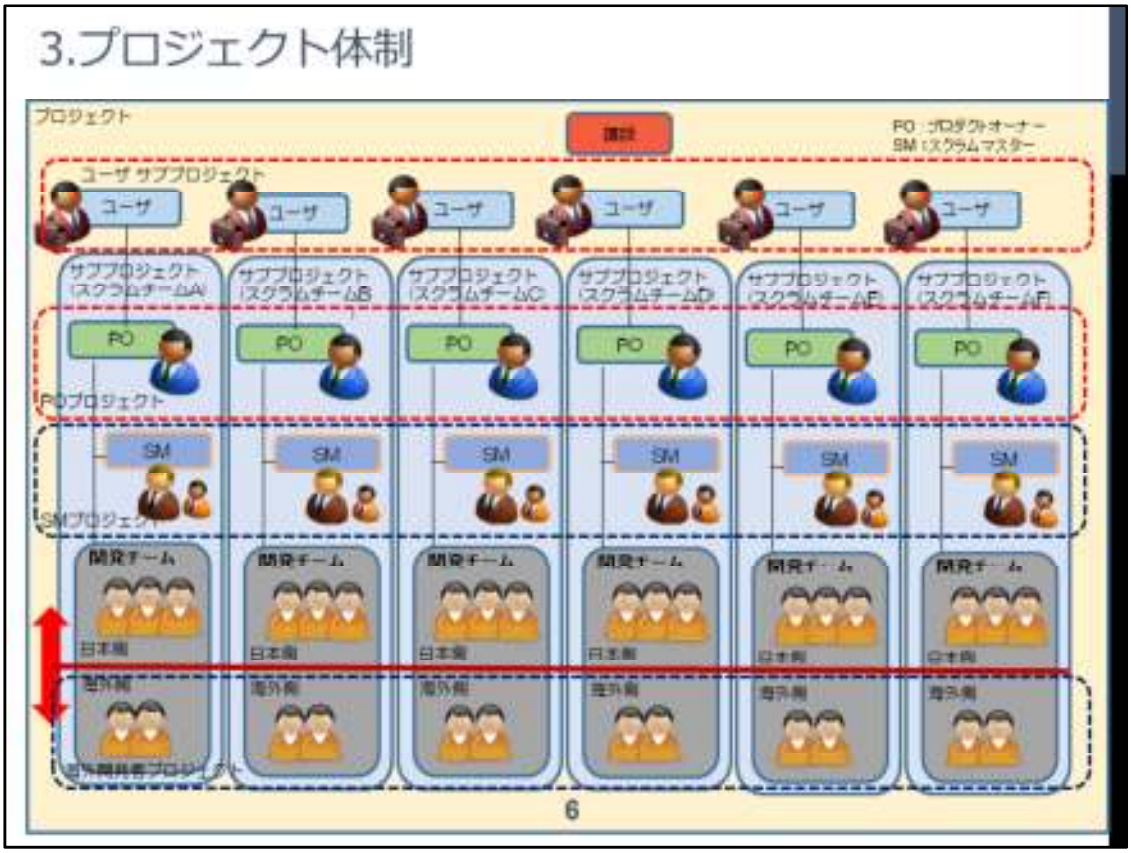
http://redmine.jp/

Redmineの機能について説明します。

Redmineには、このような機能があります。
各機能は、Redmineのタブで分かれています。

スクラムで一番使用するのが、Backlogs(かんばん)とTask board(バックログ)の機能です。

Roadmap(ロードマップ)、Gantt(ガントチャート)、Calender(カレンダー)は参考程度に使用します。
Activity(活動)は、プロジェクトメンバーのRedmineに対する日々の更新が確認できます。
News(ニュース)でプロジェクトメンバーへの伝達事項を掲載します。
Documents(文書)には教材をアップロードしておいてください。
Wikiの機能でレビュー結果を記録します。
Forums(フォーラム)は、プロジェクト内のQAや特定の話題について議論します。



プロジェクト体制について説明します。

全体統括するプロジェクトが存在し、その配下にスクラムチーム毎にサブプロジェクトが存在します。

プロダクトオーナー、スクラムマスター、海外のスクラムチームは
他のスクラムチームとの情報交換などを目的としたプロジェクトが存在します。

4.担当と役割（実証講座）

役割 ＼ チーム		スクラムチームA	スクラムチームB	スクラムチームC	スクラムチームD	スクラムチームE	スクラムチームF
ユーザ		日本の講師	日本の講師	日本の講師	日本の講師	日本の講師	企業のリーダクラス
プロダクトオーナー		日本の学生	日本の学生	日本の学生	日本の学生	日本の学生	企業の若手
スクラムマスター	メイン	日本の講師	日本の講師	日本の講師	日本の講師	日本の講師	企業のリーダクラス
	サブ	海外の講師	海外の講師	海外の講師	海外の講師	海外の講師	海外の講師
開発メンバ		日本の学生	日本の学生	日本の学生	日本の学生	日本の学生	企業の若手
		海外の学生	海外の学生	海外の学生	海外の学生	海外の学生	海外の学生

※SM(サブ)：海外の講師は海外の学生のフォロー

7

各々の役割が誰になるのかを説明します。

スクラムマスター（サブ）は海外の講師が担当します。
海外の学生とうまくコミュニケーションが取れない場合
日本の講師が、海外の講師と連携してください。

プロジェクトにおける役割について説明します。

以下のドキュメントを参考にしてください。
1.40_スクラム実習　課題説明_教師向け.pptx
1.50_スプリント実習　課題説明_教師向け.pptx

プロジェクト情報について説明します。

各スクラムチーム毎にプロジェクトが作成されています。

スクラムのサンプルプロジェクトとして実装済み機能のプロジェクトが作成されています。

各種資料をダウンロードできるサイトも作成されています。
※学生が使用する教材はこのプロジェクトに格納します。

グループ情報について説明します。

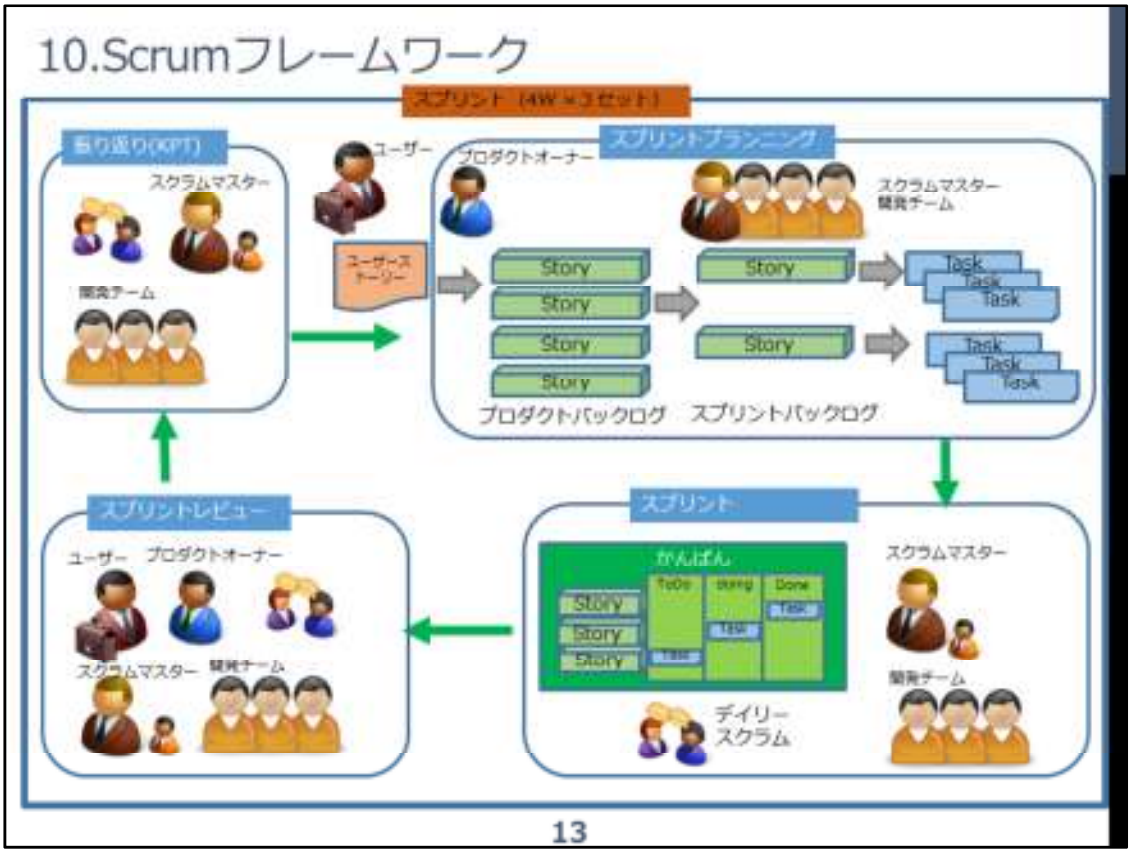
各スクラムチーム、各役割のグループが作成されています。

ユーザ情報につい説明します

初回ログイン時、パスワード、表示名を変更してください

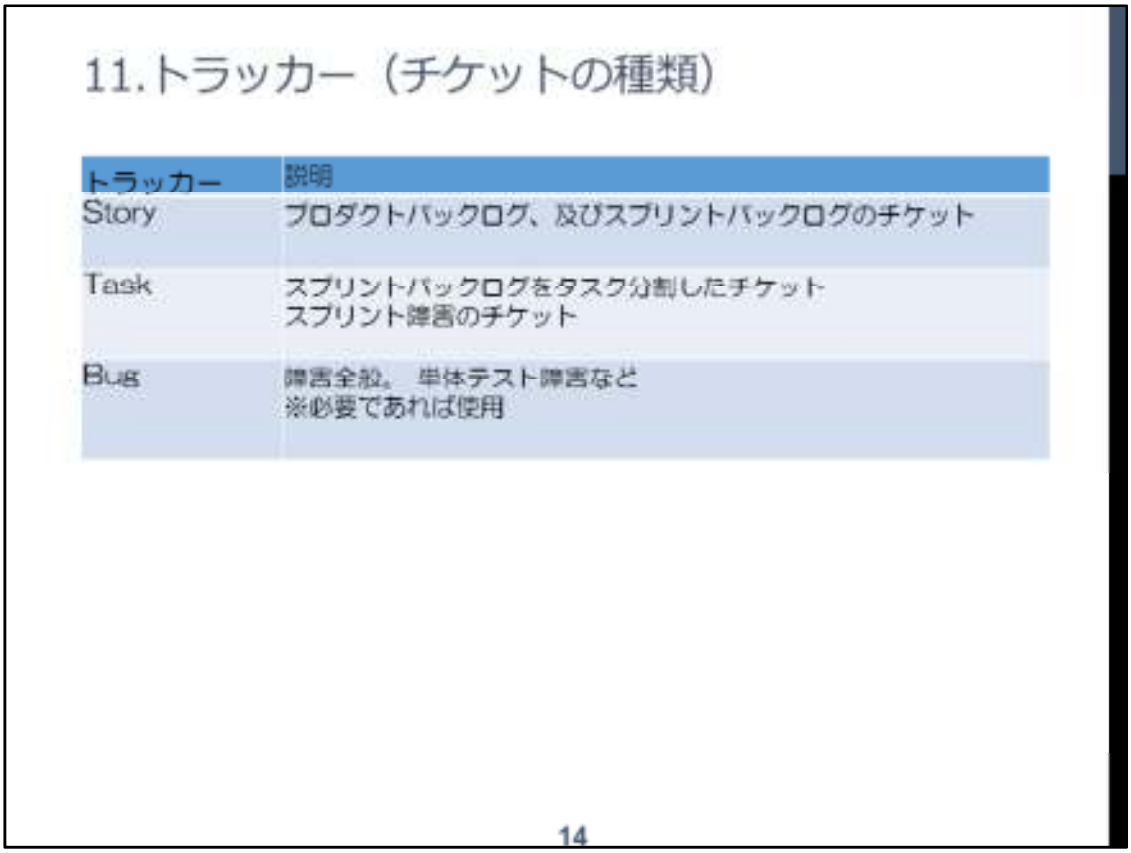
ユーザー情報の変更方法について説明します。

海外の開発者と英語でコミュニケーションとるので、表示言語は英語を推奨します。
Redmineでは、情報更新の際にメール通知する機能があります。
「参加しているプロジェクトのすべての通知」を推奨します。



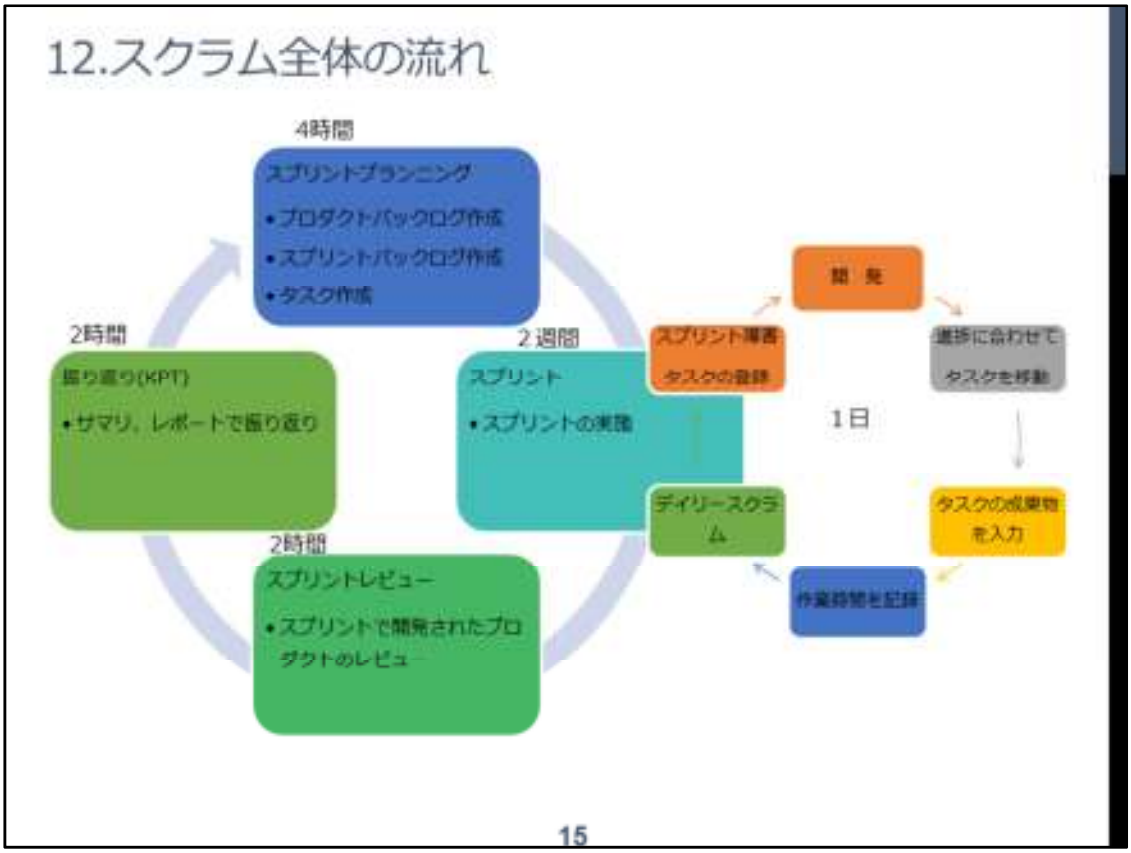
Redmineでのスクラムの流れを説明します。

1.ユーザーからユーザーストーリーが提示されます。
2.プロダクトオーナーがユーザーストーリーからプロダクトバックログを作成します。
3,スクラムマスターと開発チームがプロダクトバックログからスプリントバックログを作成します。
　また、スプリントバックログからタスクを作成します。
4.開発チームがスプリント（開発）を行います。
　　進捗に合わせてタスクを更新します。
5.毎日デイリースクラムを行います。
6.スプリント（開発）が終わったらスプリントレビューを行います。
7.その後、振り返りを行います。
8.次のスプリントを行います。



Redmineはチケットで管理しています。
チケットの種類について説明します。

Storyチケット、Taskチケットを中心に使用します。
バグの管理をする場合は、Bugチケットを使用します。



スクラム全体の流れの中で、Redmineでどのような操作をしなければならないかを説明します。

スプリントプランニング
　　プロダクトバックログ作成
　　スプリントバックログ作成
　　タスク作成
スプリント
　　スプリントの実施
　　開　発
　　進捗に合わせてタスクを移動
　　タスクの成果物を入力
　　作業時間を記録
　　デイリースクラム
　　スプリント障害
　　タスクの登録
スプリントレビュー
　　スプリントで開発されたプロダクトのレビュー
振り返り(KPT)
　　サマリ、レポートで振り返り

※次の資料ではそれぞれの操作方法が説明されています。

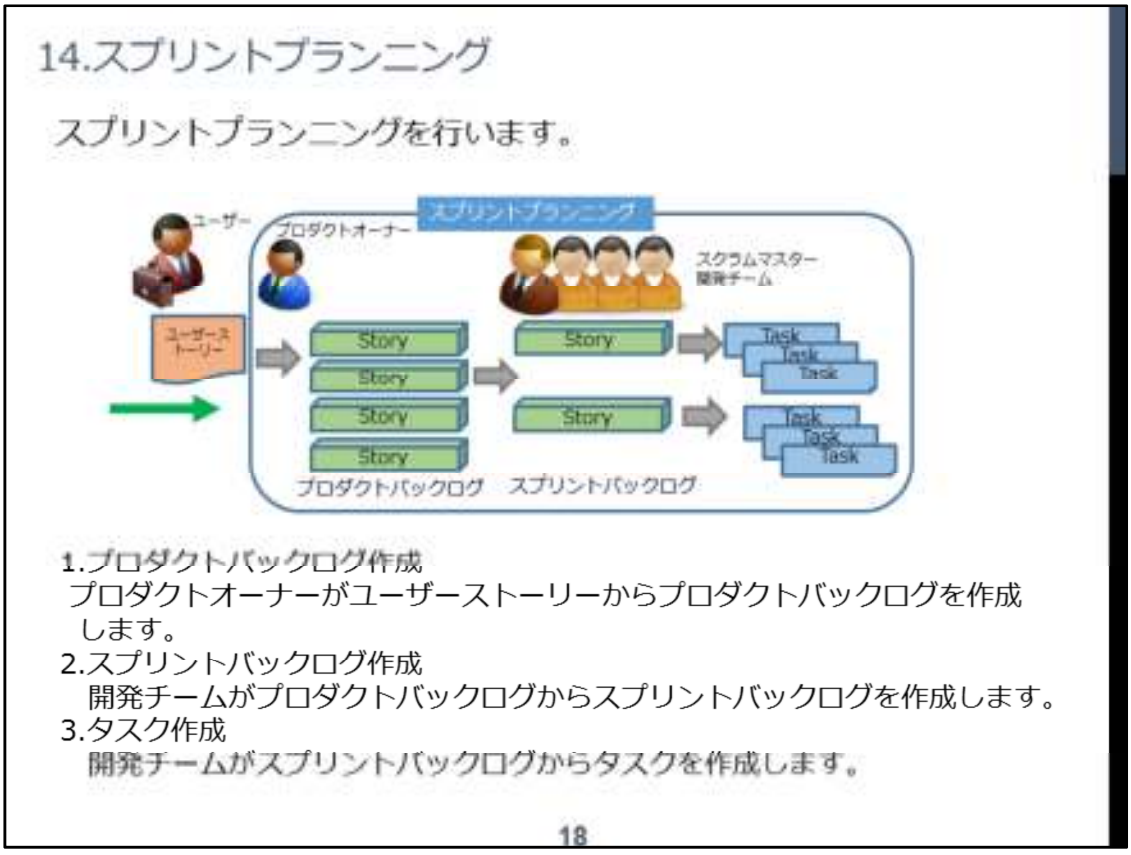
項目	内容	利用ツール	利用機能	役割					
				ユーザー	プロダクトオーナー	スクラムマスター		開発者	
						主担当	副担当	主担当	副担当
スプリントプランニング	プロダクトバックログ作成	Redmine	Backlogs(バックログ)		○				
	スプリントプランニングミーティング	Skype	会話、チャット			△	△	○	○
	スプリントバックログ作成	Redmine	Backlogs(バックログ)			△	△	○	○
	スプリントプランニング	Redmine	Taskboard（かんばん）			△	△	○	○
	タスクを作成	Redmine	Taskboard（かんばん）			△	△	○	○
スプリント	タスクの更新	Redmine	Taskboard（かんばん）			△	△	○	○
	デイリースクラム	Skype	チャット、会話			△	△	○	○
	デイリースクラムの記録	Redmine	Wiki			△	△	○	○
	作業時間の記録	Redmine	Issues（チケット）			△	△	○	○
	タスク成果物の記録	Redmine	Issues（チケット）			△	△	○	○
	スプリント障害タスクの入力	Redmine	Taskboard（かんばん）			△	△	○	○
スプリントレビュー	リリースプロダクトの格納	Redmine	Files(ファイル)			△		○	
	スプリントレビュー	Skype	チャット、会話	○	○	△	△	○	○
	スプリントレビューの結果	Redmine	Wiki			△		○	
振り返り	振り返り(KPT)	Skype	会話（チャット）			△	△	○	○
	作業時間の記録の参照	Redmine	Overview（概要）			△	△	○	
	バーンダウンチャート	Redmine	Task board（かんばん）			△	△	○	
	振り返り結果	Redmine	Wiki			△		○	

それぞれの役割の利用ツールや利用機能について説明します。
○は実施、△はアドバイス、フォローです。

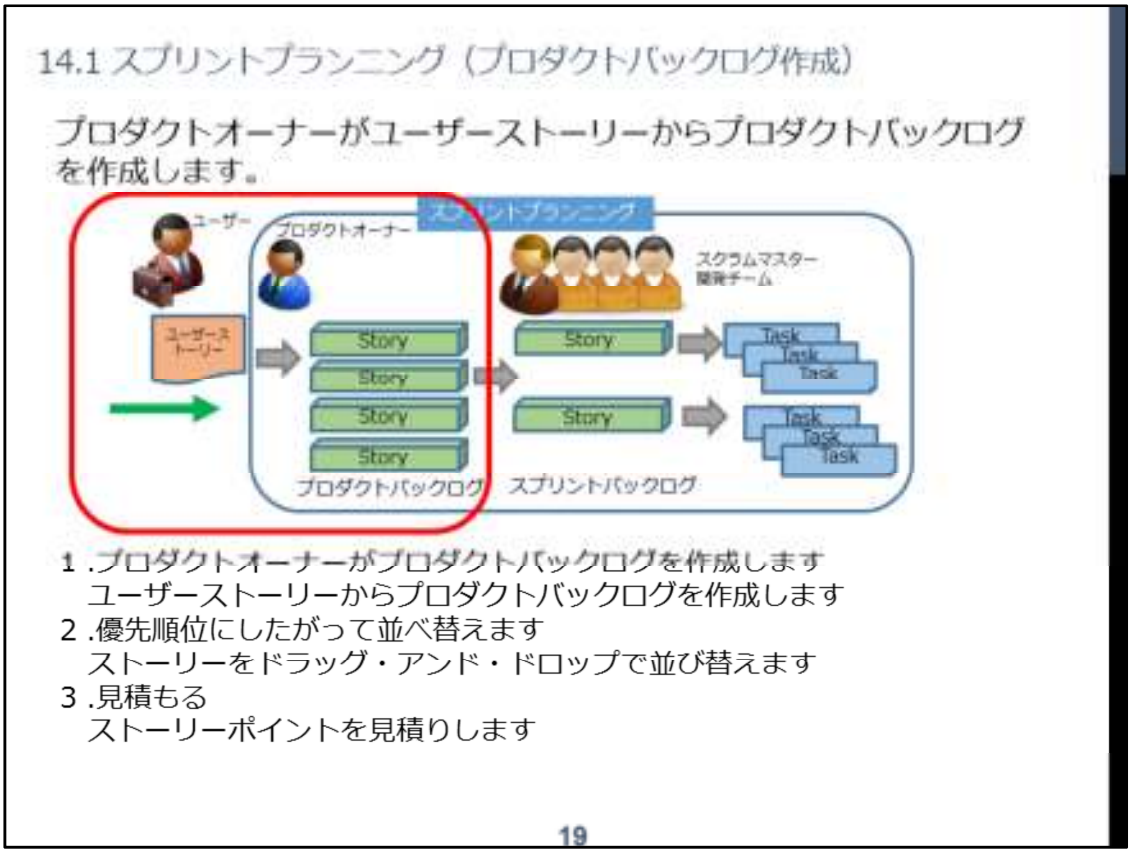
スクラムマスターはアドバイス、フォローの役割です。
あくまで開発者が主体であることを説明してください。

議題	内容	利用ツール	利用機能	リーダー	プロダクトオーナー	スクラムマスター 国内	スクラムマスター 海外	開発者 国内	開発者 海外
ＱＡやスプリント障害タスク議論	プロジェクト内、チーム内のＱＡやスプリント障害タスク議論の場	Redmine	Forums（フォーラム）		○	○	○	○	○
		Skype	会話（チャット）		○	○	○	○	○
情報交換	技術的な情報交流やその他のコミュニケーションの場	Redmine	Forums（フォーラム）		○	○	○	○	○
		Skype	会話（チャット）	○	○	○	○	○	○
情報共有	プロジェクト内、チーム内へのお知らせなどに使用	Redmine	News（ニュース）	○	○	○	○	○	○

各種ＱＡやスプリントの障害タスク議論、情報交換、情報共有は
Redmineのフォーラム機能で実施します。



スプリントプランニング方法について説明します。
1.プロダクトバックログ作成
プロダクトオーナーがユーザーストーリーからプロダクトバックログを作成します。
2.スプリントバックログ作成
開発チームがプロダクトバックログからスプリントバックログを作成します。
3.タスク作成
開発チームがスプリントバックログからタスクを作成します。



プロダクトバックログ作成方法について説明します。
プロダクトオーナーがユーザーストーリーからプロダクトバックログを作成します。

1 .プロダクトオーナーがプロダクトバックログを作成します
ユーザーストーリーからプロダクトバックログを作成します
2 .優先順位にしたがって並べ替えます
ストーリーをドラッグ・アンド・ドロップで並び替えます
3 .見積もる
ストーリーポイントを見積りします

14.1 スプリントプランニング（プロダクトバックログ作成）1

プロダクトマスターがユーザーストーリーからプロダクトバックログを作成します

1.プロジェクトメニューから「Backlogs(バックログ)」を選択します

2.Product Backlog(プロダクトバックログ)から「New Story(新しいストーリー)」を選択します

20

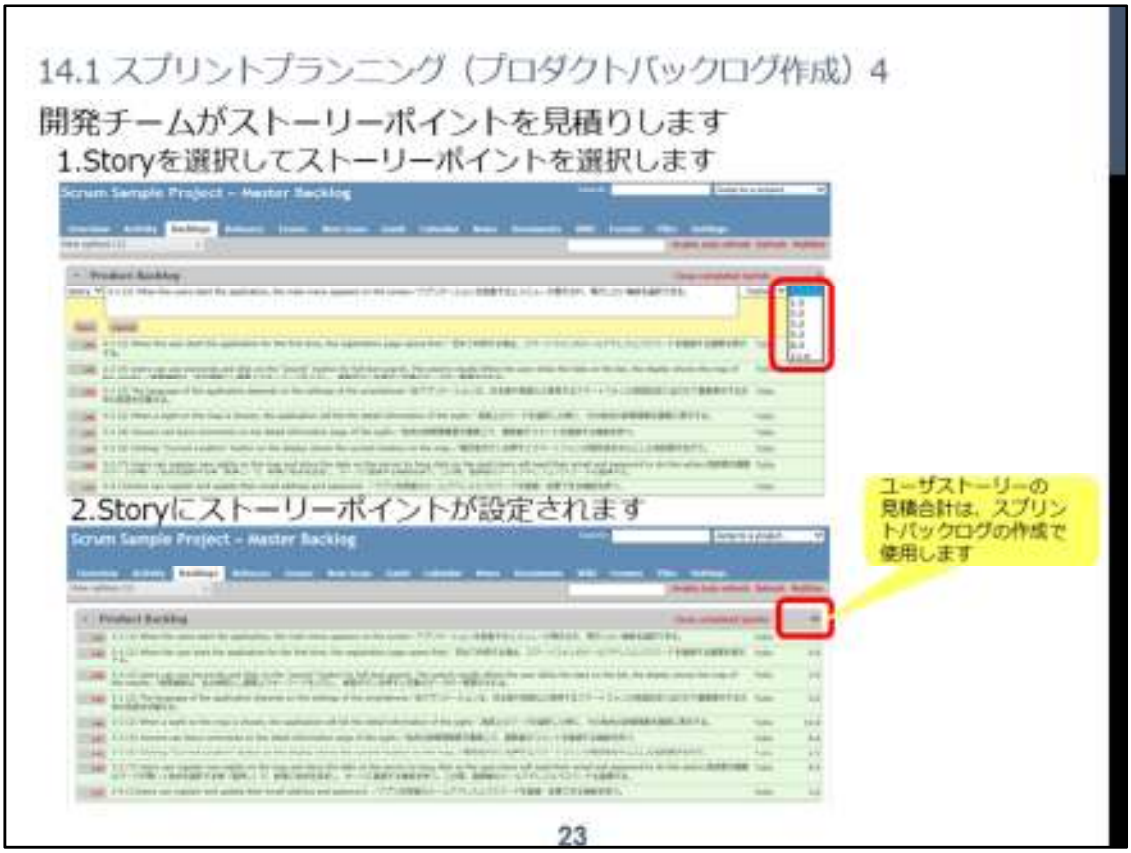
プロダクトバックログ作成方法について説明します。

プロダクトバックログ作成方法について説明します。

プロダクトオーナーは、ユーザーストーリーからすべてのプロダクトバックログを作成します。
海外開発者が理解できるように日本語と英語で記載します。

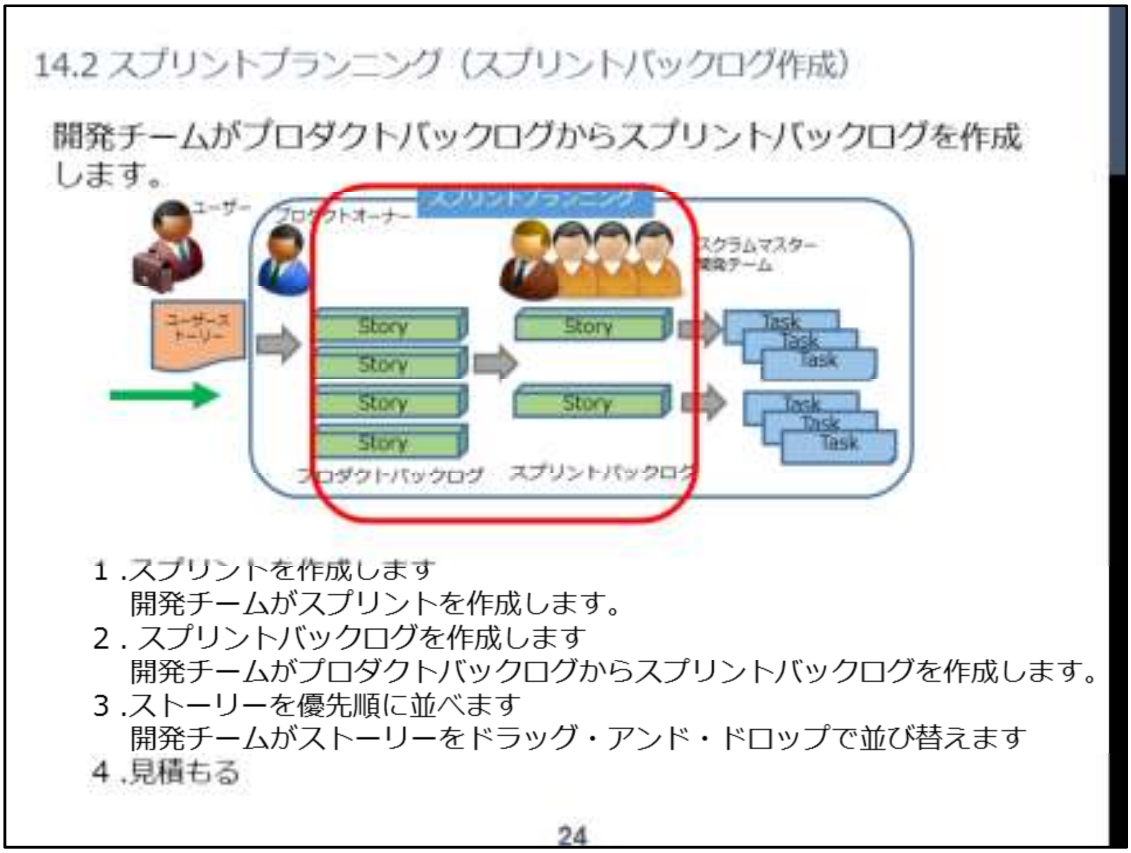
プロダクトバックログ作成方法について説明します。

プロダクトオーナーが優先順位にしたがってストーリーを並べ替えます
※優先順位はプロダクトオーナーが判断します。



プロダクトバックログ作成方法について説明します。

開発チームがストーリーポイントを見積りします
見積は、プランニングポーカーなどを使用します
ストーリーポイントの見積合計は、スプリントバックログの見積で使用します。



スプリントバックログ作成方法について説明します。
開発チームがプロダクトバックログからスプリントバックログを作成します。

1 .スプリントを作成します
　開発チームがスプリントを作成します。
2 . スプリントバックログを作成します
　開発チームがプロダクトバックログからスプリントバックログを作成します。
3 .ストーリーを優先順に並べます
　開発チームがストーリーをドラッグ・アンド・ドロップで並び替えます
4 .見積もる

14.2 スプリントプランニング（スプリントバックログ作成）1

開発チームがプロダクトバックログからスプリントを作成します。

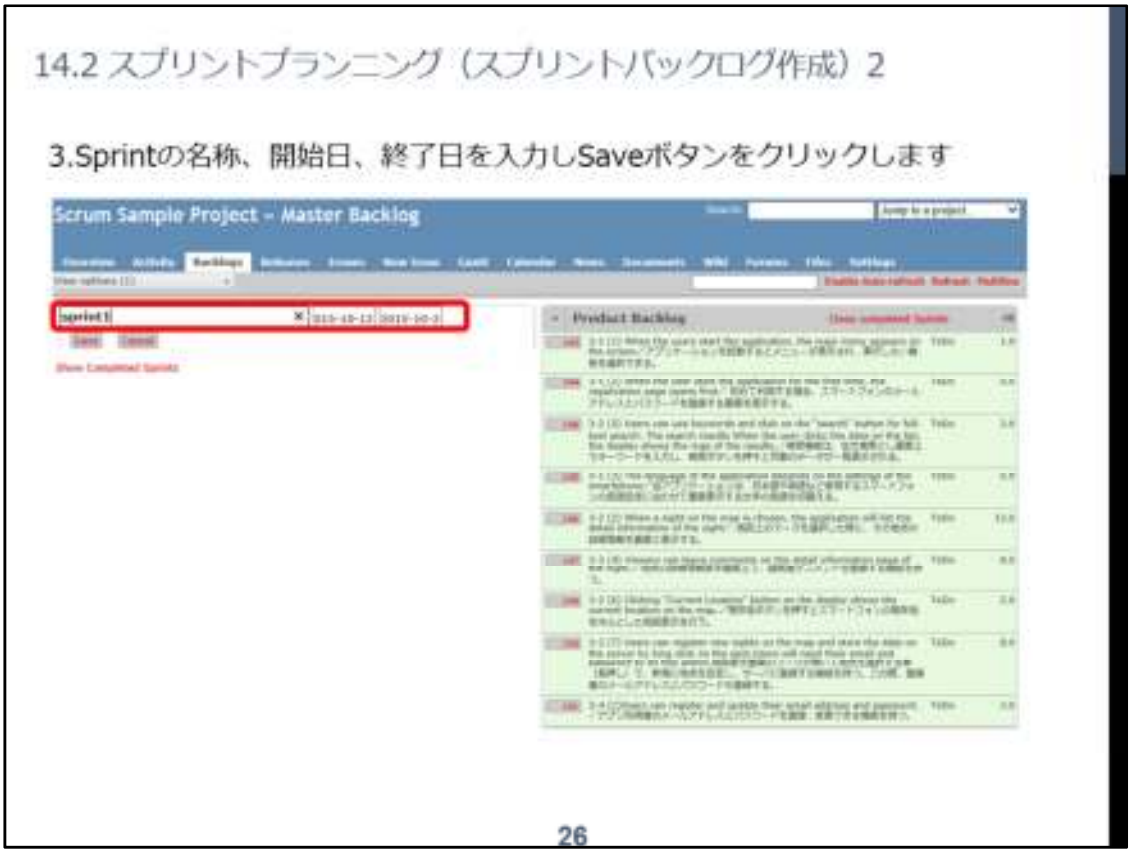
1.プロジェクトメニューから「Backlogs(かんばん)」を選択し、
「New Sprint (新しいスプリント) 」を選択します

※Multilineをクリックすると、複数行で表示できます。

25

スプリントバックログ作成方法について説明します。

開発チームがプロダクトバックログからスプリントを作成します。

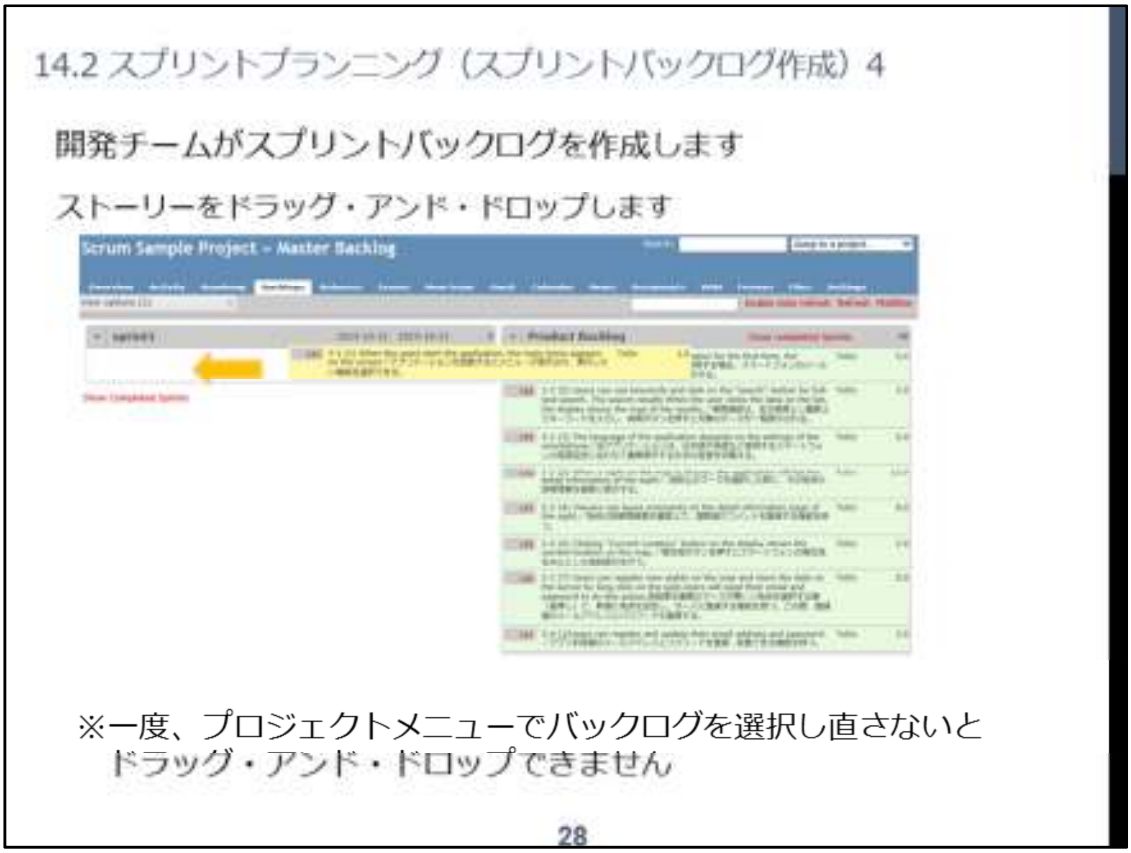


スプリントバックログ作成方法について説明します。

新しいスプリントにSprintの名称、開始日、終了日を入力します。

スプリントバックログ作成方法について説明します。

新しいSprintが登録されます



プロダクトバックログをスプリントバックログに移動させる方法を説明します。

開発チームがスプリントバックログを作成します

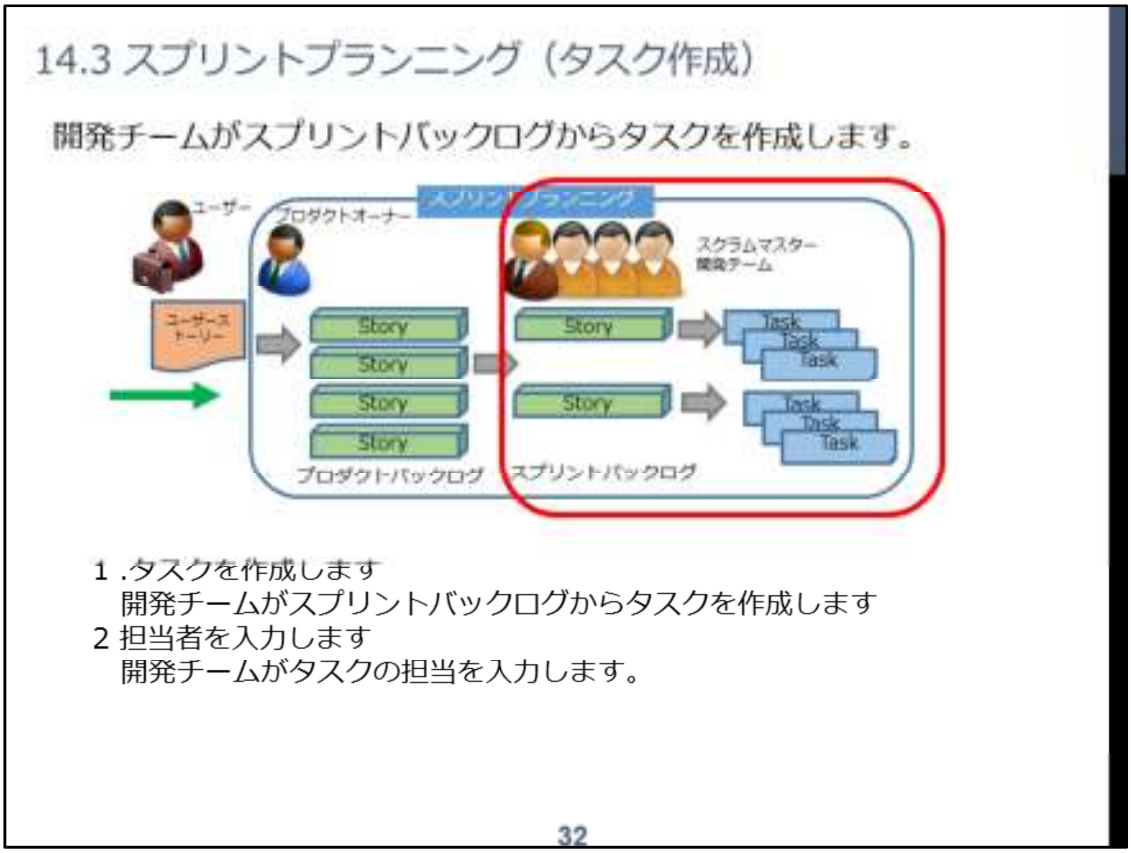
プロダクトバックログを１回のスプリントでどれだけ実施するかを説明します。

すべての作業が３スプリントで完了できることを想定していますが、
スキル等を考慮して適宜変更してください。

スプリントバックログの優先順の並び替え方法について説明します。

開発チームがスプリントバックログを優先順をつけて並び替えます
※スプリント内の優先順は開発チームが判断します

スプリントバックログが並び替わったことを説明します。



タスクの作成方法について説明します
開発チームがスプリントバックログからタスクを作成します。

1 .タスクを作成します
開発チームがスプリントバックログからタスクを作成します
2 担当者を入力します
開発チームがタスクの担当を入力します。

14.3 スプリントプランニング（タスク作成）1

開発チームがタスクを作成します
Taskboard（かんばん）を選択します

タスクの作成方法について説明します
対象スプリントののTaskboard（かんばん）を選択します

対象スプリントのTaskboard（かんばん）が表示されることを説明します。

新しいタスクを作成する方法を説明します。

Subject(題名)には、作業内容（調査、設計、実装、テストなど）を入力します。

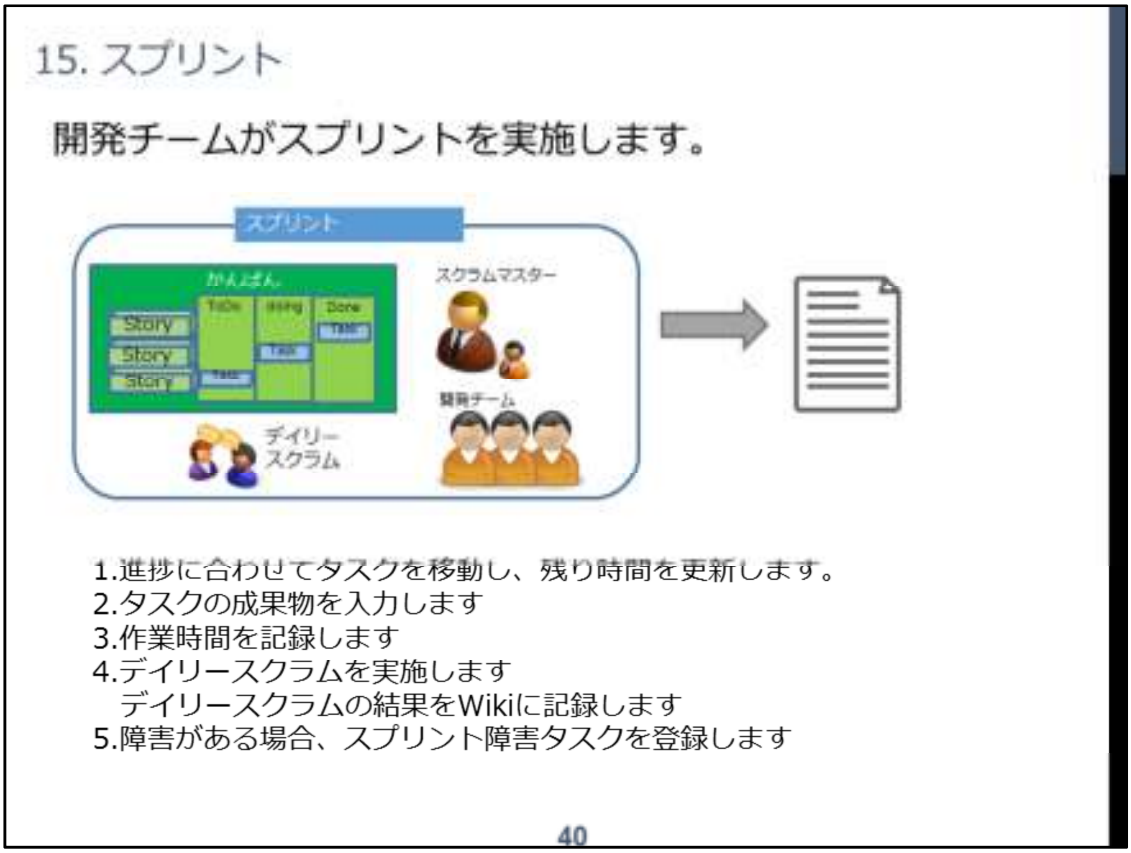
Remaining (hours)（残り時間）には、見積時間を入力します。
教材では１２時間以下としていますが、適宜変更しても構いません。

新しいタスクを作成されたことを説明します。

スプリントバックログのタスクを作成することを説明します。

開発者がタスクにサインアップして担当者を入力することを説明します。

担当者を入力することで色が変わることを説明します。

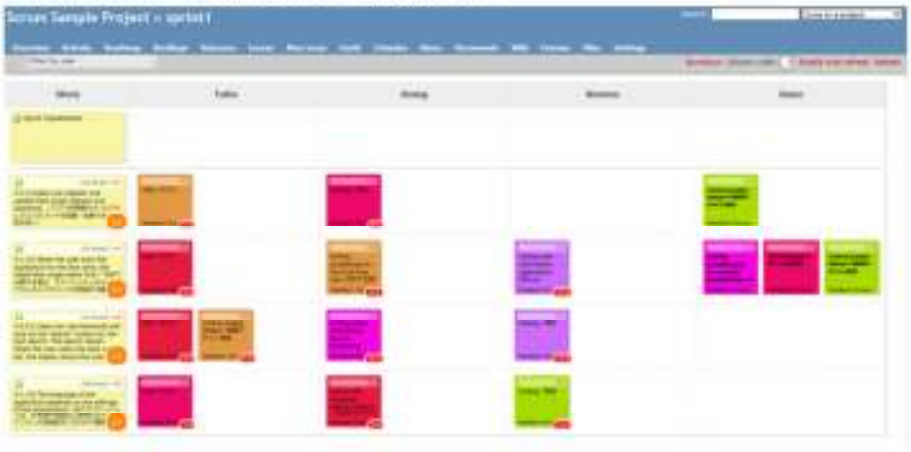


スプリント時の操作方法について説明します
開発チームがスプリントを実施します。

1.進捗に合わせてタスクを移動し、残り時間を更新します。
2.タスクの成果物を入力します
3.作業時間を記録します
4.デイリースクラムを実施します
　デイリースクラムの結果をWikiに記録します
5.障害がある場合、スプリント障害タスクを登録します

15.1 スプリント（タスクの更新）

進捗にあわせて随時タスクを移動します



ステータス	説明
ToDo	作業前（やること）のチケット
Doing	進行中（仕掛中）のチケット
Review	レビューするチケット　※担当者以外がレビューし、レビュー完了後Doneに移動
Done	完了したチケット

※各タスクの成果物はタスクチケットに記載します。

41

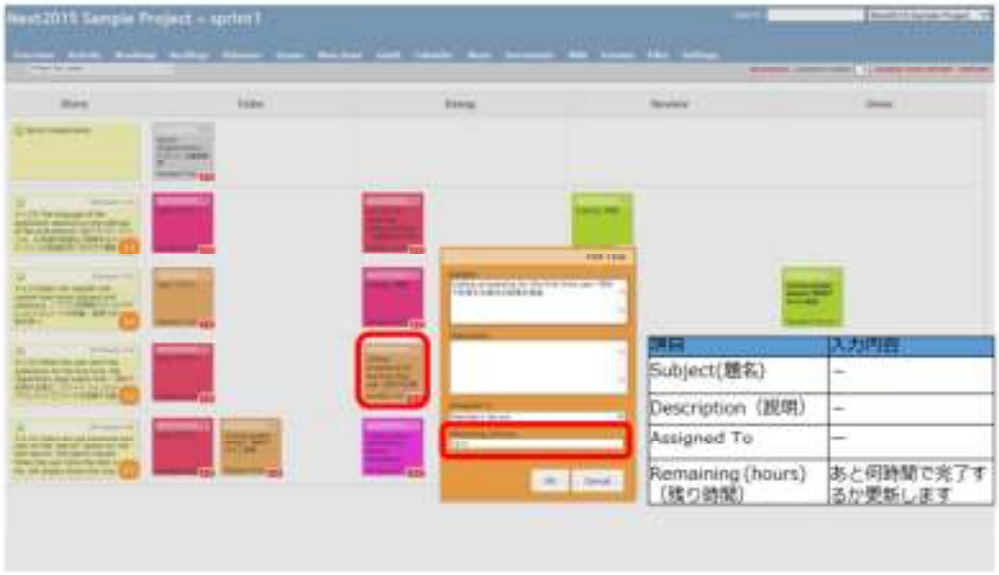
タスクの更新の方法について説明します。

担当者は進捗にあわせて随時タスクを移動し、完了させます。

Review状態になったら、担当者以外が成果物を確認しDoneに移動します。

15.1 スプリント（タスクの更新）1

タスクがあと何時間で完了するか、Remaining (hours)（残り時間)を更新します。

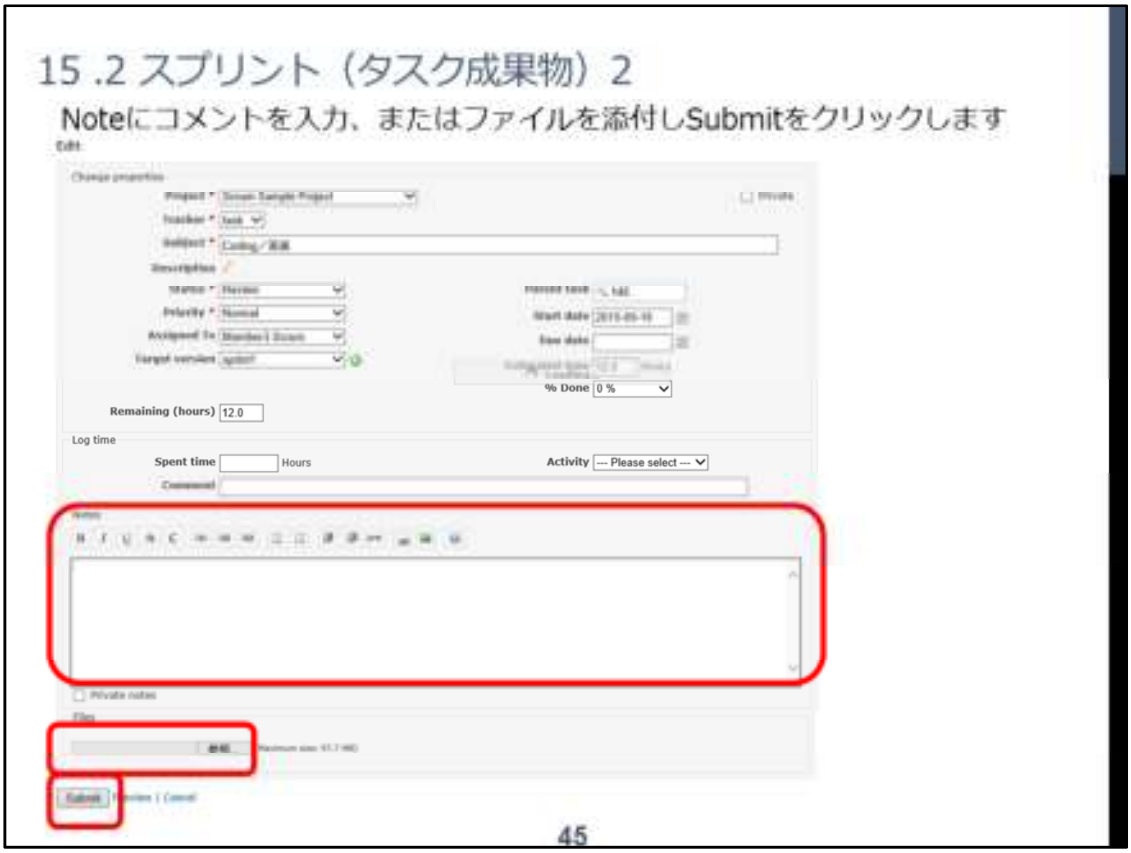


42

Remaining (hours)（残り時間）には、あと何時間で完了するかを入力します

タスクの成果物を入力する方法を説明します。

task画面で対象となるチケットを選択し、編集します。



TaskチケットのNoteにコメントを入力し、成果物を添付します。
成果物は設計メモ、ソースコード、テスト結果などを想定しています。

15.2 スプリント（デイリースクラム）

開発チームがデイリースクラムを行い、結果をWikiに記録します。
全員で進捗を確認するミーティングです。
「昨日やったこと」、「今日やること」、「障害になっていること」を
共有します。

※国内開発者は英語と日本語で記載します

46

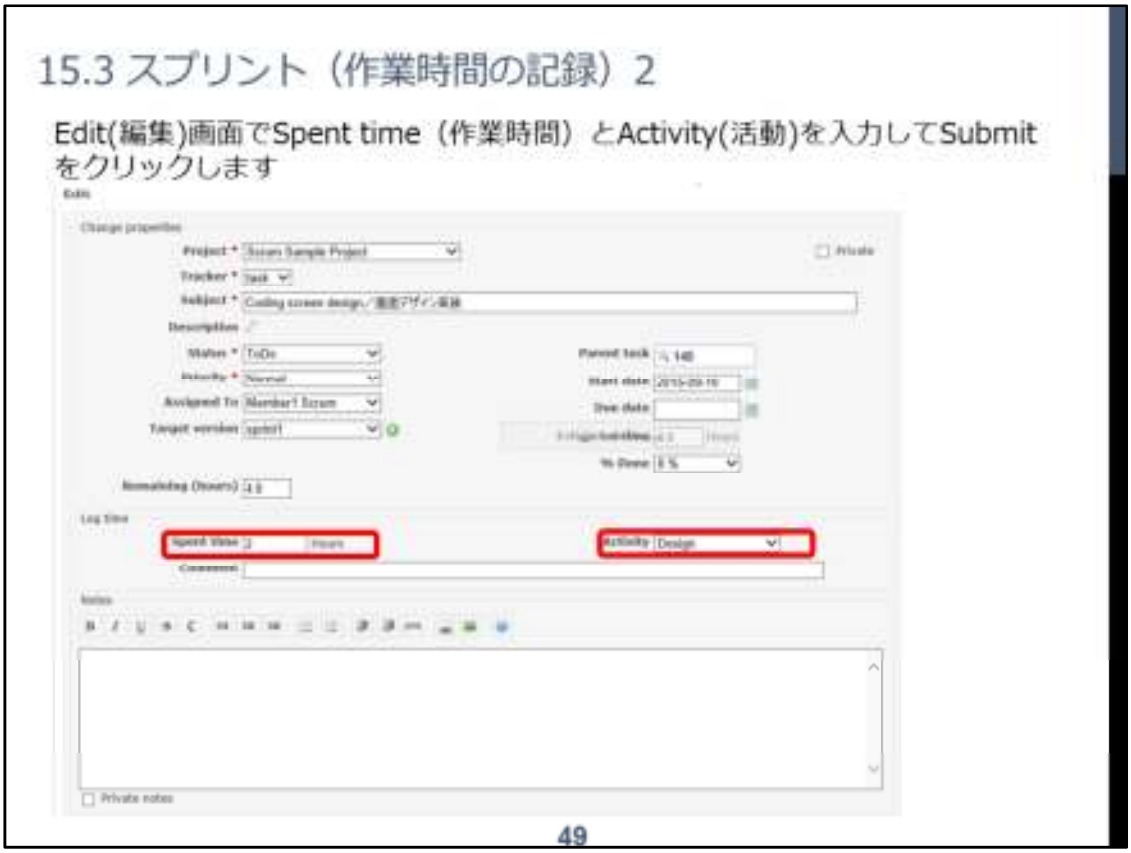
開発チームのデイリースクラムの結果をWikiに記録します。

通常は、ミーティングを行った結果を記載しますが、
海外開発チームとの会話でのコミュニケーションが難しいため、
ディリースクラムを始める前に記載しておくことを推奨します。

作業時間を記録する方法を説明します。

通常、スクラムでは作業実績は記録しませんが、次のスクラムの見積のためにも
実際行った作業時間を記録します。

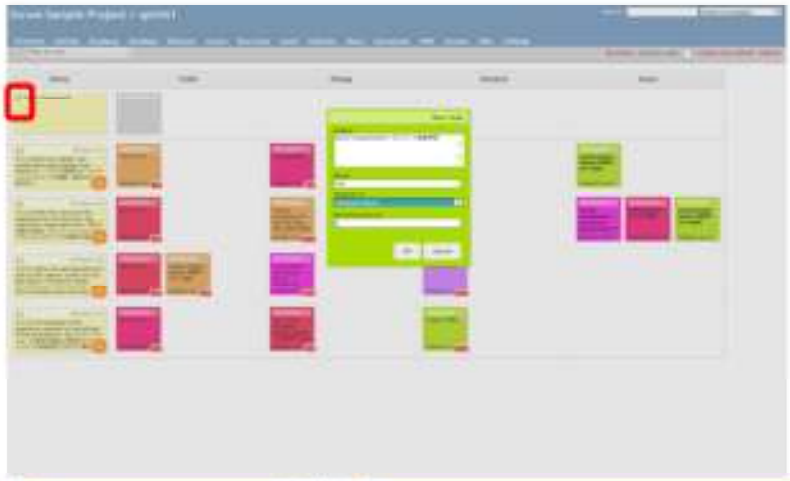
作業時間を記録する方法を説明します。



作業時間を記録する方法を説明します。

15.4 スプリント（スプリント障害タスクの入力）

スクラムで発生した障害を入力します
Sprint Impediments（スプリント障害事項）の「+」をクリックします

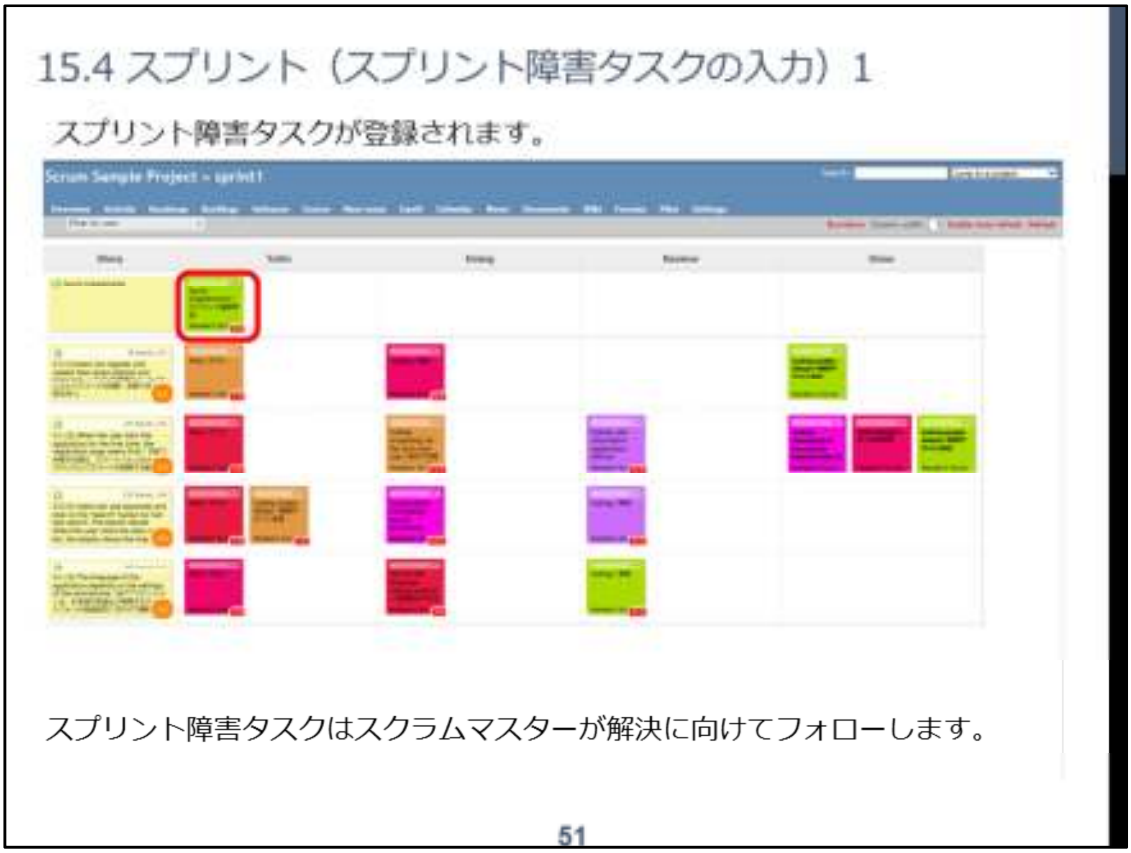


項目	入力内容
Subject(題名)	題名を入力します。（英語／日本語）
Blocks	障害となっているタスクNoを入力します
Assigned To	担当者を入力します
Remaining (hours)（残り時間）	見積時間を入力します

50

スクラムで発生した障害はスプリント障害タスクを作成します。

スプリント障害タスクの担当者は、タスクを作成した担当者にします。



スプリント障害タスクの担当者は、スクラムマスターが解決に向けてフォローします。

15.5 スプリント（バーンダウンチャート）

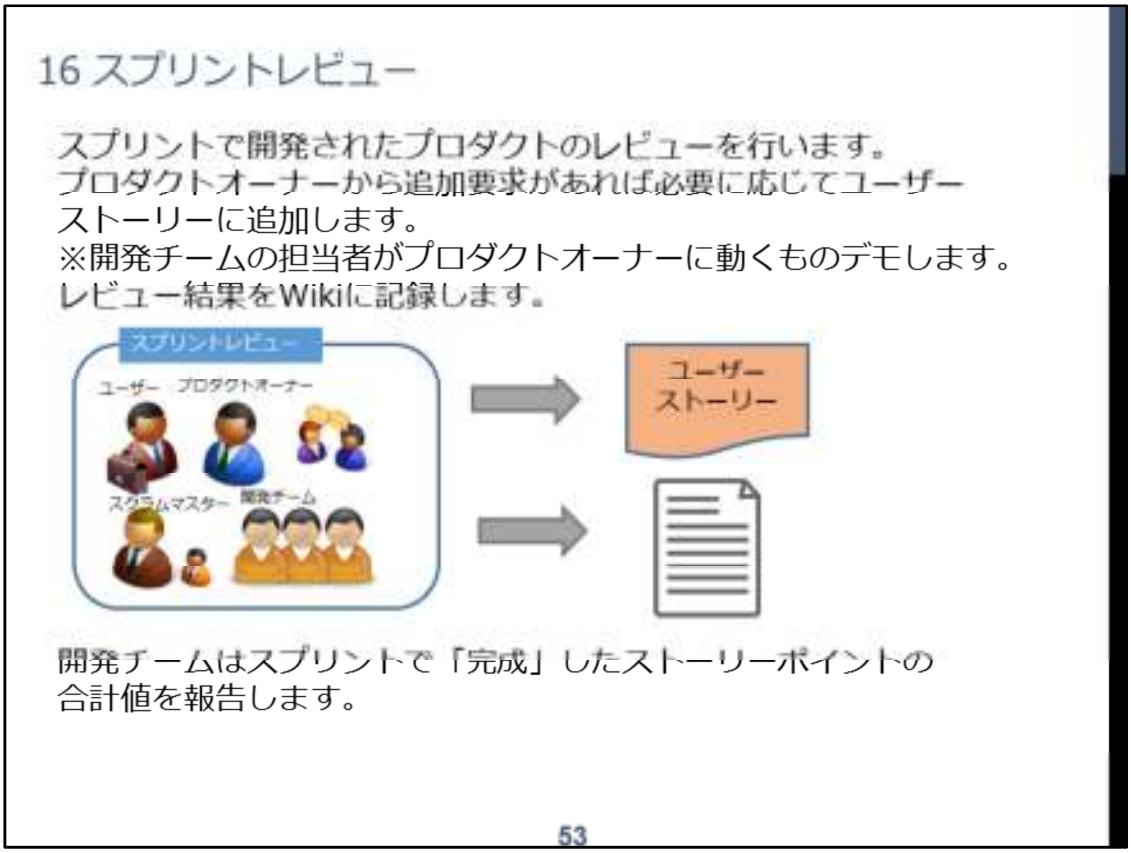
Task board（かんばん）のBourndown(バーンダウン)ボタンをクリックするとバーンダウンチャートが表示されます。

項目名（英語）	項目名（日本語）
ideal	必要なバーンレート(ポイント)
points committed	必要なバーンレート(時間)
points not accepted	承認されたポイント
points not resolved	残り時間
remaining hours	理想時間
required burn rate (hours)	費やすべきポイント
required burn rate (points)	配分すべきポイント

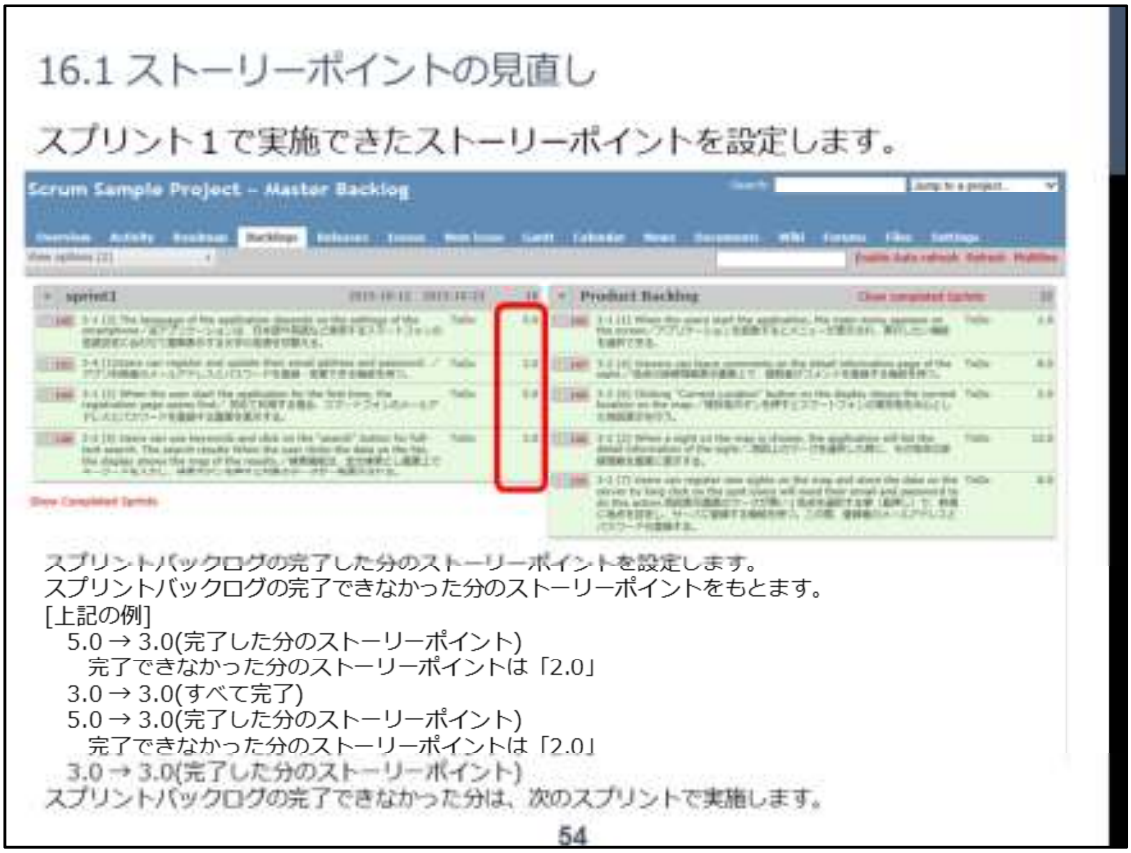
※完了までにあとどれくらい時間が必要かを確認できます

52

タスクの完了までにあとどれくらい時間が必要かをバーンダウンチャートで確認します。



スプリントで開発されたプロダクトのレビューを行います。
プロダクトオーナーから追加要求があれば必要に応じてユーザーストーリーに追加します。
開発チームの担当者がプロダクトオーナーに動くものデモします。
レビュー結果をWikiに記録します。
開発チームはスプリントで「完成」したストーリーポイントの合計値を報告します。



スプリントレビュー時にストーリーポイントの見直しを行います。

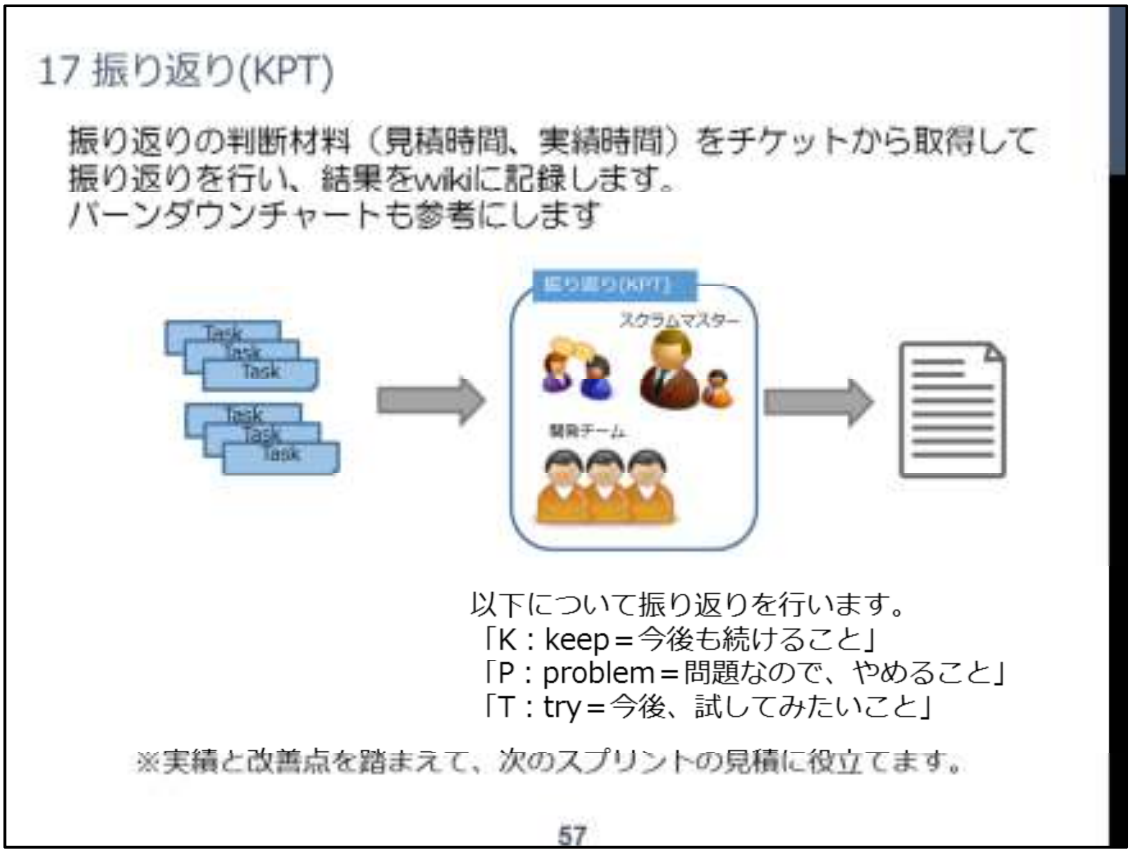
完了した分のストーリーポイントを設定し、完了できなかった分のストーリーポイントを求めます。

完了できなかった分のストーリーポイントをプロダクトバックログに追加します。

完了した分のストーリーポイントがベロシティになります。

スプリントレビューの結果を、Wikiに記録します。

国内の開発者は、英語と日本語で記載します。



振り返りの方法を説明します。

振り返りの判断材料（見積時間、実績時間）をチケットから取得して
振り返りを行い、結果をwikiに記録します。
バーンダウンチャートも参考にします

予定作業時間と作業実績時間の差を確認し
作業時間に乖離があれば、原因を分析します。

スプリント時に入力した作業時間が参照できます。

作業実績時間を参照する方法を説明します。

振り返りの結果を、Wikiで情報共有します。

各人が振り返りを行い、最後にチームとしての見解をまとめてください。

国内開発者は英語と日本語で記載します

Redmine-Backlogs サーバの構築手順書

目次

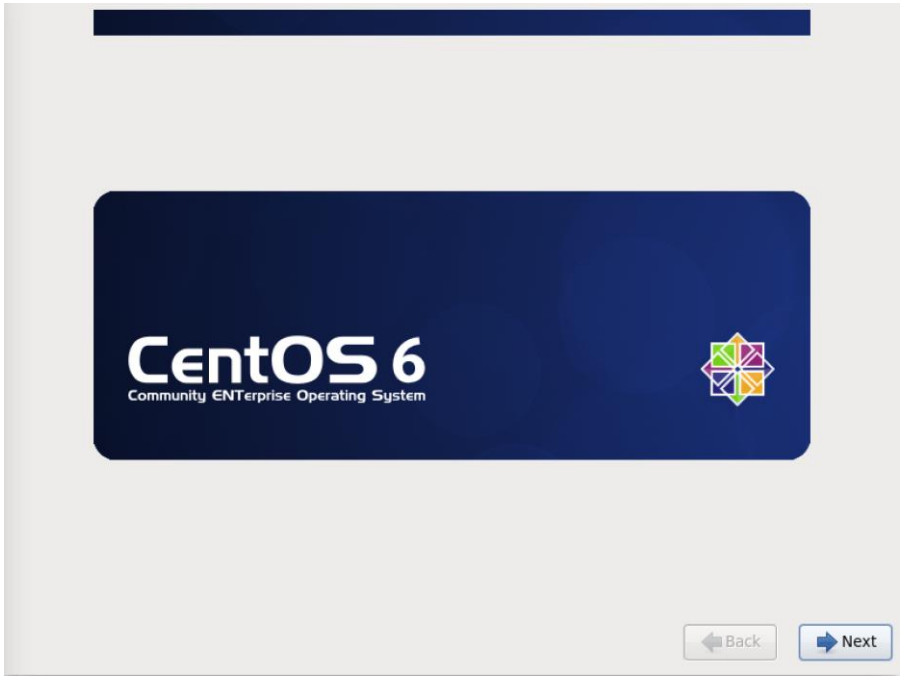
I. Redmine—Backlogs サーバのインストール 2
(1) CentOS のインストール 2
(2) Linux への開発ツールインストール 3
(3) bitnami-redmine のダウンロード 3
(4) bitnami-redmine のインストール 3
(5) Redmine インストール後の初期設定 4
(6) Holidays Gem(Plug-in)のインストール 5
(7) iCalendar Gem(Plug-in)のインストール 5
(8) Prawn Gem(Plug-in)のインストール 5
(9) Redmine の自動起動設定 5
(10) Backlogs プラグインのダウンロード 6
(11) Backlogs プラグインのインストール 6
(12) nokogiri プラグインのインストール 6
(13) DB のマイグレーションの実施 6
(14) サーバの再起動とトラッカー登録 7
(15) ステータス登録 7
(16) 新規トラッカーのワークフロー登録 8
(17) Backlogs プラグインのインストール 9
(18) Redmine の再起動 9
(19) Redmine サーバの動作確認 9
II. Redmine—Backlogs サーバの初期設定について 11
(1) ログイン画面の英語化とアカウント自動登録禁止設定 11
(2) カリキュラムで使用するプロジェクトの新規作成 13
(3) 教材を使用するユーザの登録 14
(4) その他の追加情報など 16

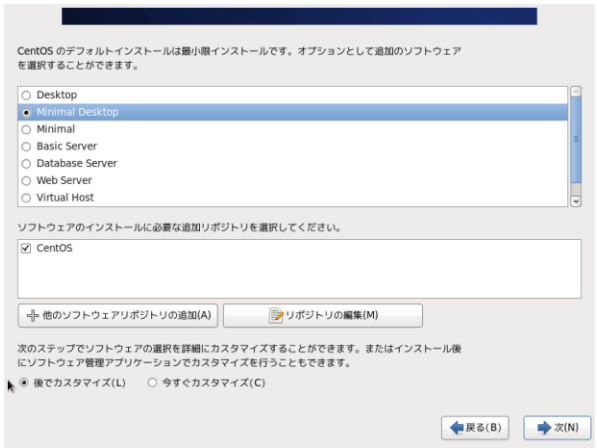
I. Redmine-Backlogs サーバのインストール

以下に Linux OS の 1 種である CentOS(Ver6.6) 64Bit 上に Redmine 及び Backlogs プラグインをインストールする場合の手順を示す。今回のプロジェクトで利用する環境は、Redmine プロジェクト管理サーバに、スクラム開発手法を効率良く管理する為の Backlogs プラグインを追加して構築する。

(1) CentOS のインストール

CentOS Ver6.6 64Bit 版を規定の手順に従ってインストールする。インターネットへの接続が正しく実施できる様に環境を構築する。その際、“Minimam Desktop”環境として構築する。また、redmine ユーザも作成しておき、sudo コマンドを利用できるように設定しておく。





(2) Linux への開発ツールインストール

OS インストール完了後、root ユーザでログインして開発ツールのインストールを下記のコマンドにて行う。

→ yum groupinsatall “Development Tools”

また Git は別途インストールされるので、一度下記のコマンドにてアンインストールしておく。

→ yum erase git

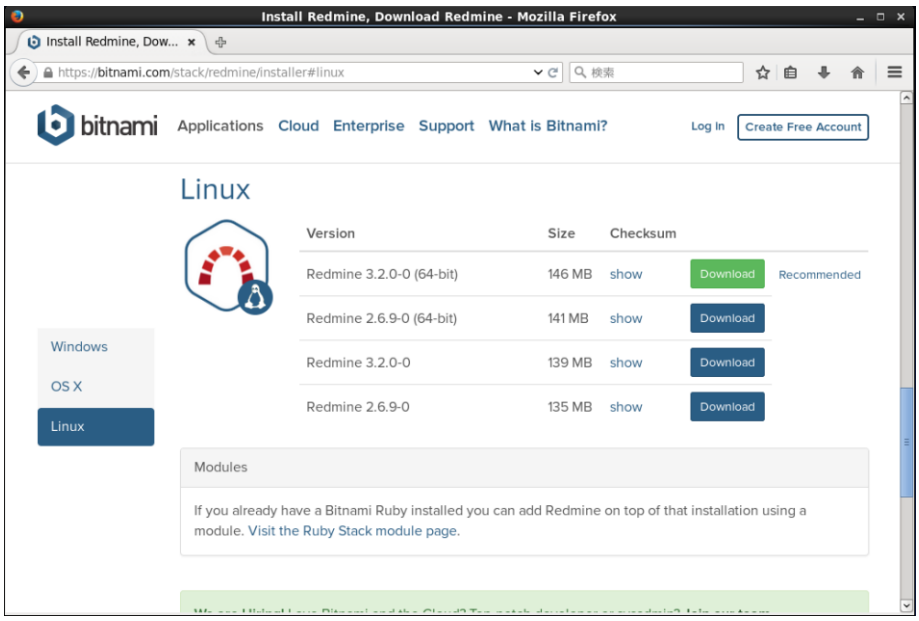
上記インストール終了後に、OS 全体のアップデートを実施しておく。

→ yum update

(3) bitnami-redmine のダウンロード

redmine ユーザでログインして、Bitnami の HP より、bitnami-redmine(ここでは v2.6.9 64bit 版を対象)をダウンロードする。URL は下記の通り。

https://bitnami.com/stack/redmine/installer#linux



※ これ以降の作業は、全て redmine ユーザにて実施する。

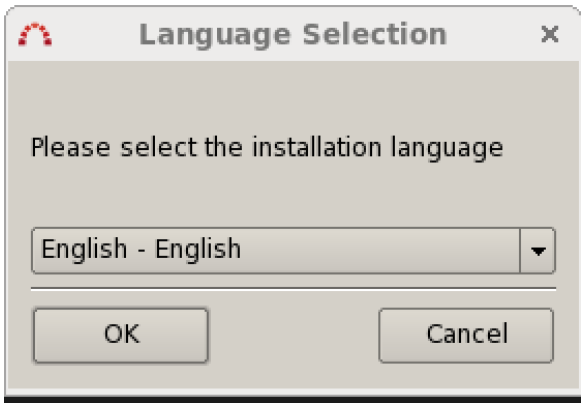
(4) bitnami-redmine のインストール

以下のコマンドを実行し、Bitnami-redmine をインストールする。

→ chmod 755 bitnami-redmine-2.6.9-0-linux-x64-installer.run

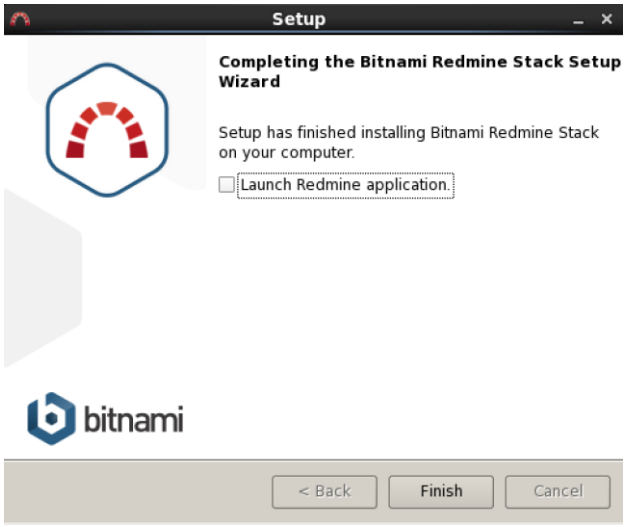
→ sudo ./ bitnami-redmine-2.6.9-0-linux-x64-installer.run

※ 言語＝英語でインストールを実行する。



※ インストラーの問いに、デフォルトで答えて進めていく。

※ Admin ユーザとして、redadmin を登録し、デフォルト言語＝英語、インストール時最後の問合せで、自動起動のチェックを外しておく。



(5) Redmine インストール後の初期設定

以下のコマンドを実行して、Redmine の実行環境に切替える。

→ cd /opt/redmine-2.6.9-0

→ ./use_redmine

→ cd /opt/redmine-2.6.9-0/apps/redmine/htdocs

→ bundle install –without development test

上記コマンド実行時にエラーが出ても無視する。

(6) Holidays Gem(Plug-in)のインストール

以下のコマンドを実行して、holidays Gem(２バージョン)をインストールする。

→ gem install holidays –version 1.0.3

→ gem install holidays

(7) iCalendar Gem(Plug-in)のインストール

以下のコマンドを実行して、icalendar Gem をインストールする。

→ gem install icalendar

(8) Prawn Gem(Plug-in)のインストール

以下のコマンドを実行して、prawn Gem をインストールする。

→ gem install prawn

(9) Redmine の自動起動設定

以下のコマンドを実行して、自動実行用スクリプトをコピーする。

→ sudo cp /opt/redmine-2.6.9-0/ctlscript.sh /etc/init.d/bitnami-redmine

コピーしたスクリプトをテキストエディタで開き、以下のように編集する。

エディターの実行： sudo vi /etc/init.d/bitnami-redmine

以下のテキスト行をスクリプトの先頭に追加する。

```
#! /bin/sh
### BEGIN INIT INFO
# Prvides:    Bitnami Redmine
# Required-Start:      $remote_fs $syslog
# Required-Stop:       $remote_fs $syslog
# Default-Start:       2345
# Default-Stop:        0 1 6
# Short-Description: Start daemon at boot time
# Description: Enable service provided by daemon.
### END INIT INFO
```

上記行を追加後、下記のコマンドを実行し自動起動の設定を完了する。

→ sudo chkconfig –add /etc/init.d/bitnami-redmine

→ sudo chkconfig bitnami-redmine on

(10) Backlogs プラグインのダウンロード

plugins ディレクトリに移動して、Git コマンドを使用して Backlogs をダウンロードする。

→ cd /opt/redmine-2.6.9-0/apps/redmine/htdocs/plugins

→ git clone git://github.com/backlogs/redmine_backlogs.git

→ cd redmine_backlogs

→ git checkout v1.0.6

(11) Backlogs プラグインのインストール

以下のコマンドを実行して、Backlogs プラグインのマイグレーションを実施する。

→ export RAILS_ENV=production

→ cd /opt/redmine-2.6.9-0/apps/redmine/htdocs

→ bundle exec rake db:migrate

※ 上記コマンド実行時にエラー表示され、bundle install を実施する様に指示された場合、下記のコマンドを実行する。

→ bundle install –no-deployment

※ またエラーが発生した場合、Gemfile 内の下記の行を編集してから実施する必要がある。

→ vi /opt/redmine-2.6.9-0/apps/redmine/htdocs/plugins/redmine_backlogs/Gemfile

gem "nokogiri", "< 1.6.0"

↓ ↓

gem "nokogiri" <- 不必要なバージョン指定を外す

(12) nokogiri プラグインのインストール

→ gem install nokogiri --use-system-libraries=true --with-xml2-include=/usr/include/libxml2

再度、下記コマンドを実行しインストールを実施する。

→ bundle install –no-deployment

(13) DB のマイグレーションの実施

下記コマンドを実行する。

→ bundle exec rake db:migrate

→ bundle exec rake tmp:cache:clear

→ bundle exec rake tmp:sessions:clear

ここまでで、Redmine+Backlogs プラグインのインストールが完了する。

(14) サーバの再起動とトラッカー登録

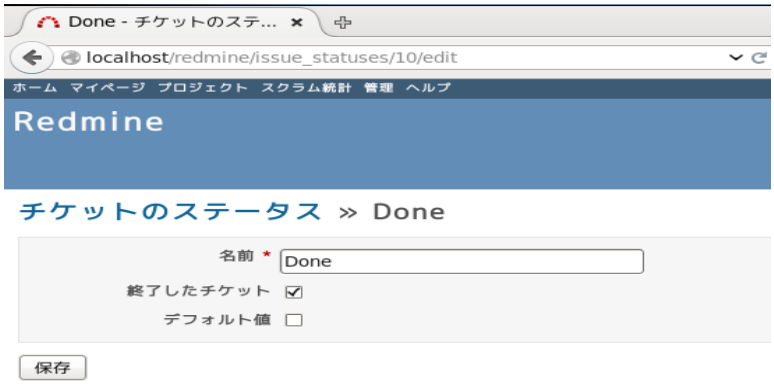
一度、サーバを再起動し、再起動後に Web ブラウザでアクセス(http://IP アドレス/redmine/)して、redadmin ユーザにてログインする。

ログイン後、トラッカー登録画面にて、story と task の 2 トラッカーを新規登録する。

(管理メニュー → トラッカー画面にて、“新しいトラッカーを作成”をクリック)

(15) ステータス登録

管理メニュー → チケットのステータス画面に移動し、“新しいステータス”ボタンより、“ToDo”、“Doing”、“Review”及び、“Done”を追加する。“Done”を追加する際、”終了したチケット“チェックボックスにチェックを入れておく。



また新規追加したステータスのソート順を上位に移動する。ToDo、Doin、Review、Doing の順番に並べておく。

(16) 新規トラッカーのワークフロー登録

上記で追加した story と task トラッカーにワークフローを設定する作業を行う。

管理メニュー → ワークフロー画面に移動し、ロール＝“すべて”、トラッカー＝“story”を選択後、編集ボタンをクリックする。

表示された画面にて、ステータスの遷移先を指定する。

“task”についても同様に下記の様に設定する。

(17) Backlogs プラグインのインストール

以下のコマンドを実行してプラグインのインストールを行う。

→ cd /opt/redmine-2.6.9-0/apps/redmine/htdocs

→ bundle exec rake redmine:backlogs:install

※上記インストールの途中で、”story”と”task”トラッカーを指定する。

(18) Redmine の再起動

インストール作業がすべて完了したので、Redmine サーバの再起動を行う。

Linux OS ごと再起動するか、下記のコマンドにて Redmine サーバのみの再起動を実施してもよい。

→ service bitnami-redmine restart

(19) Redmine サーバの動作確認

再起動後に、クライアント PC の Web ブラウザより Redmine サーバにアクセスし、正しくログイン出来る事を確認する。

URI= http://IP アドレス/redmine/

10

II. Redmine−Backlogs サーバの初期設定について

この章では、インストール直後に Redmine サーバを当カリキュラムで使用する為の初期設定を行う。設定する内容は下記の各項目となる。

(1) ログイン画面の英語化（デフォルト）とアカウント自動登録の禁止

(2) カリキュラムで使用するプロジェクトの新規作成

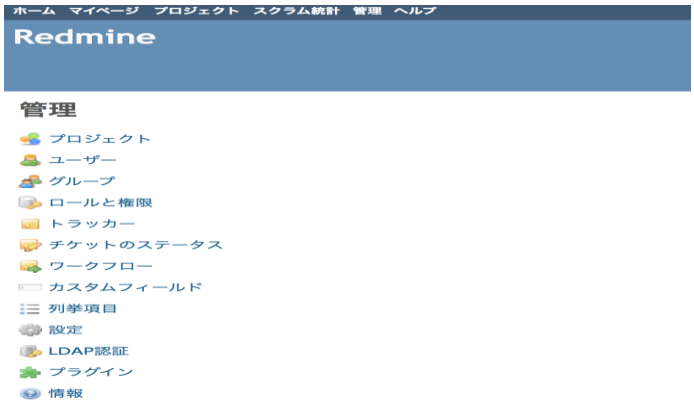
(3) 教材を利用するユーザの登録

※当カリキュラムで Redmine を使用する際の詳細なサーバ設定については、Redmine 環境設定(Backlogs).pptx を参照。

(1) ログイン画面の英語化とアカウント自動登録禁止設定

ログイン画面の英語化

1.管理者ユーザ(redadmin)でログインし、トップメニュー“管理”→“設定”を開く。



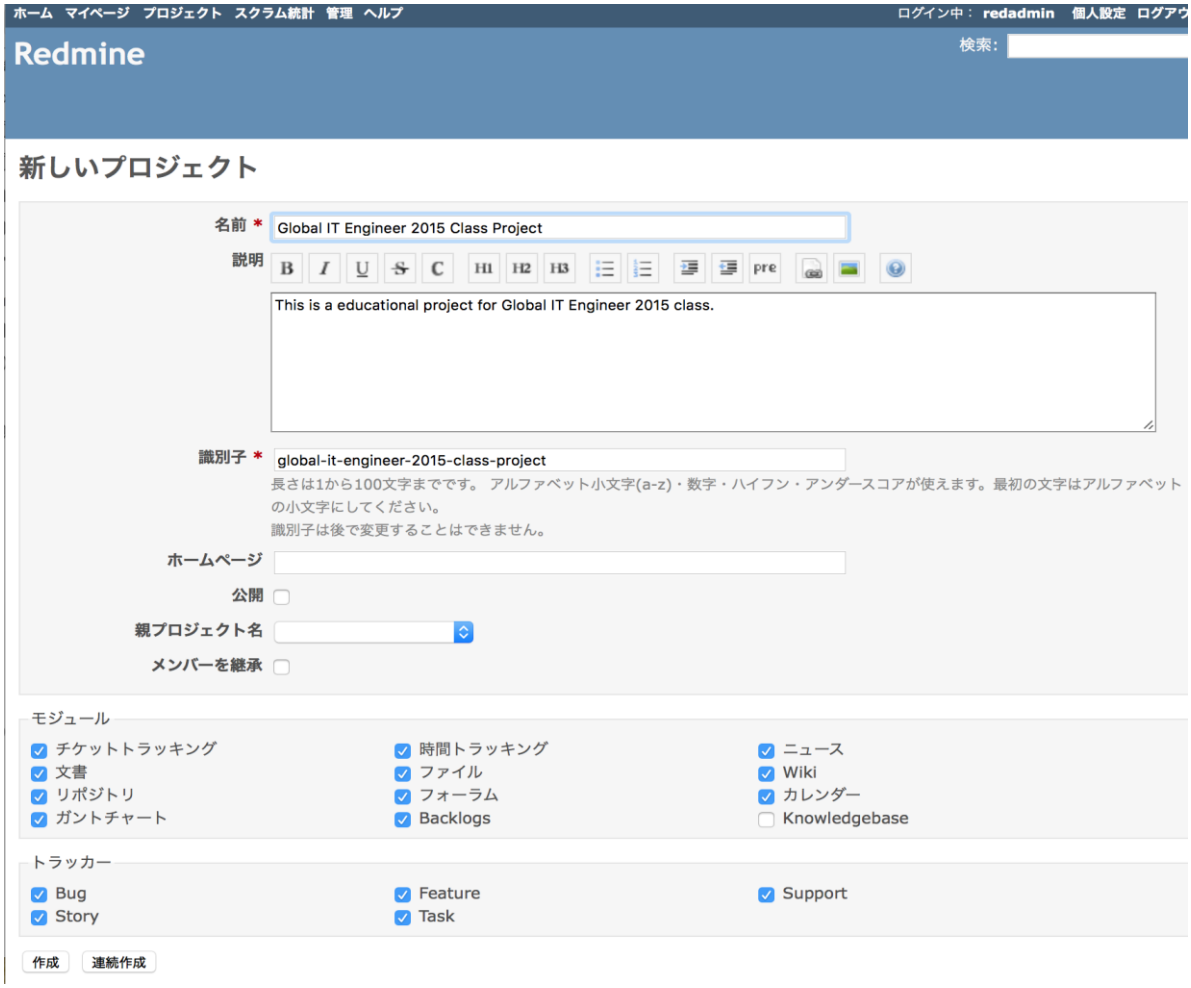
2.“設定”→”表示”タブを開き、規定の言語＝”English”を選択し、次の”匿名ユーザに既定の言語を強制”にチェックをつけて、設定を保存する。

3.次にアカウント自動登録を禁止する為に、”認証”タブをクリックし、”ユーザによるアカウント登録”＝”無効”を選択し、保存ボタンをクリックする。

(2) カリキュラムで使用するプロジェクトの新規作成

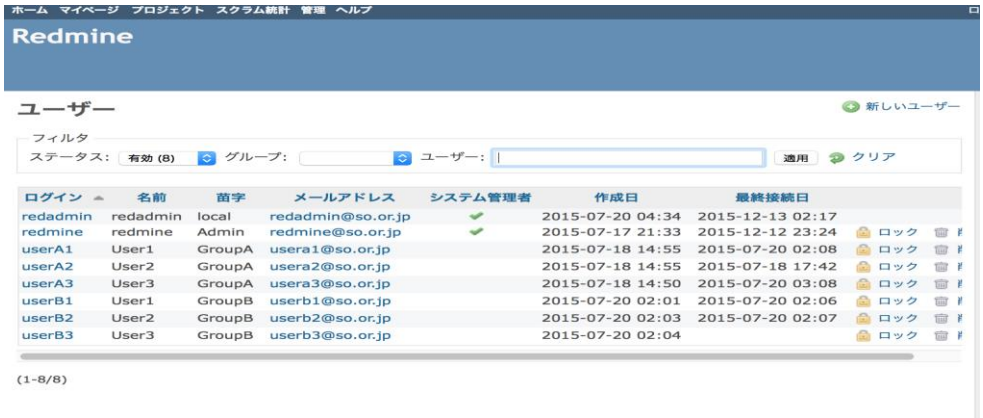
1.新しいプロジェクトを作成する為に、トップメニューの”プロジェクト”をクリックし、次に”新しいプロジェクト”をクリックする。

2.次画面にて、当カリキュラムで使用するプロジェクトの諸情報を入力し、“作成”ボタンをクリックして、新規プロジェクトを作成する。
この際に、必ずモジュール内の“Backlogs”にチェックが入っていることを確認する。



(3) 教材を使用するユーザの登録

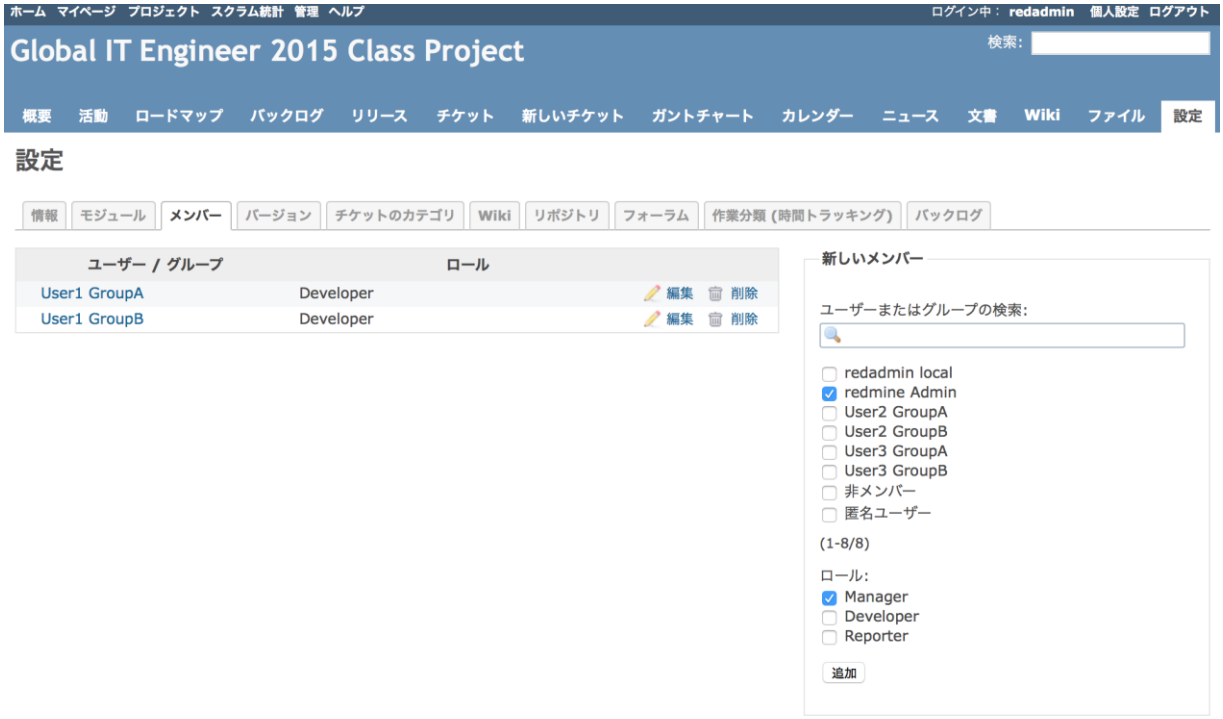
1.ユーザを作成する為に、トップメニュー“管理”→“ユーザー”を開く。次に右上の“新しいユーザー”をクリックし、ユーザーの新規登録画面を開く。



2.必要なユーザー情報を入力し、“保存”ボタンを押して登録を完了する。

3.新規登録したユーザーをプロジェクトに追加するために、トップメニュー“管理”→“プロジェクト”を開き、対象のプロジェクト名をクリックする。

4.プロジェクト設定画面で、“メンバー”タブをクリックして、ユーザーをメンバー登録する。メンバー登録する際は、対象のユーザーの権限を予め決めておく。権限は、Manager、Developer、Reporter の３種より決める。Manager は当プロジェクトに対する設定変更の権限(主に講師)を持つ、Developer は一般ユーザ向けの権限(学生)で、Reporter は参照のみの権限(外部の監視者)となっている。



右側のユーザー名とロールを選択して、“追加”ボタンを押す事でユーザーを登録できる。また当画面より、指定した権限の変更やプロジェクトからの除外なども設定できる。

※ メンバー一覧内の“非メンバー”とは、当該プロジェクトにメンバー登録されていない他のメンバーに対する権限を付与する時に利用出来る。

(4) その他の追加情報など

Redmine に関する使用方法などは、下記のサイトを参照。

http://redmine.jp/tech_note/first-step/user/

Backlogs プラグインに関する情報は、下記のサイトを参照。

http://www.redminebacklogs.net

以　上

16

地方型グローバルIT人材の育成 PBLカリキュラム
Redmine（BackLog）の環境設定

目次

はじめに …… 3

1.Redmineの設定 …… 4

2.プロジェクトの作成 …… 14

3.グループの作成 …… 20

4.ロールと権限の作成 …… 25

5.ユーザーの作成 …… 35

6.トラッカーの作成 …… 44

7.チケットのステータス作成 …… 49

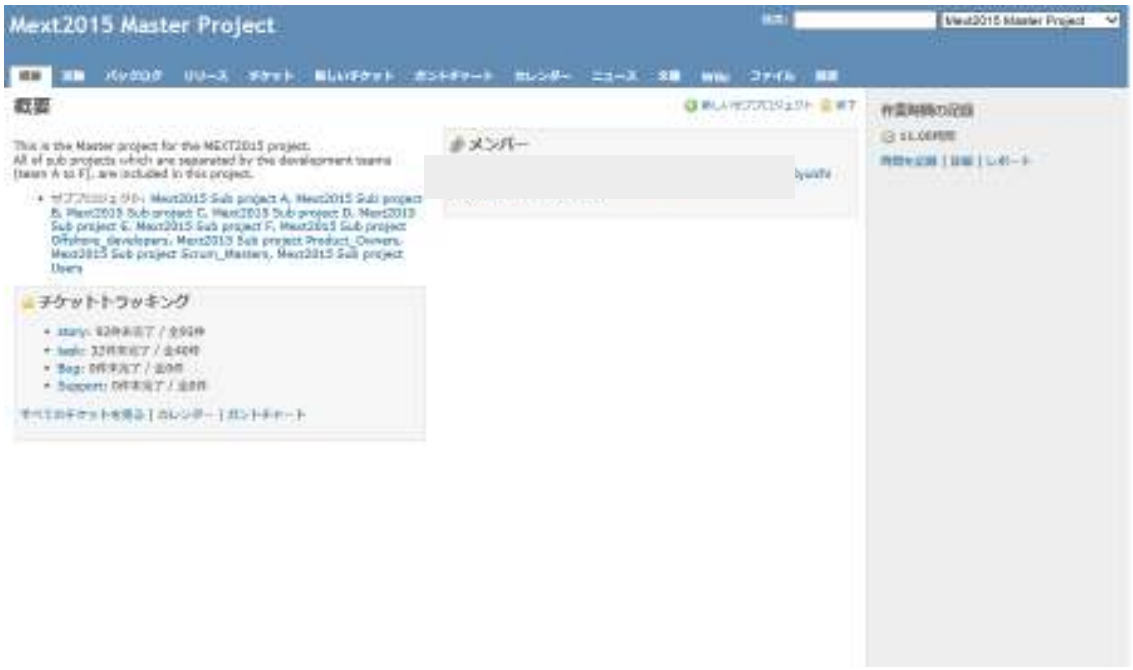
8.ワークフローの作成 …… 54

9.列挙項目の作成 …… 57

10.プラグインの設定 …… 60

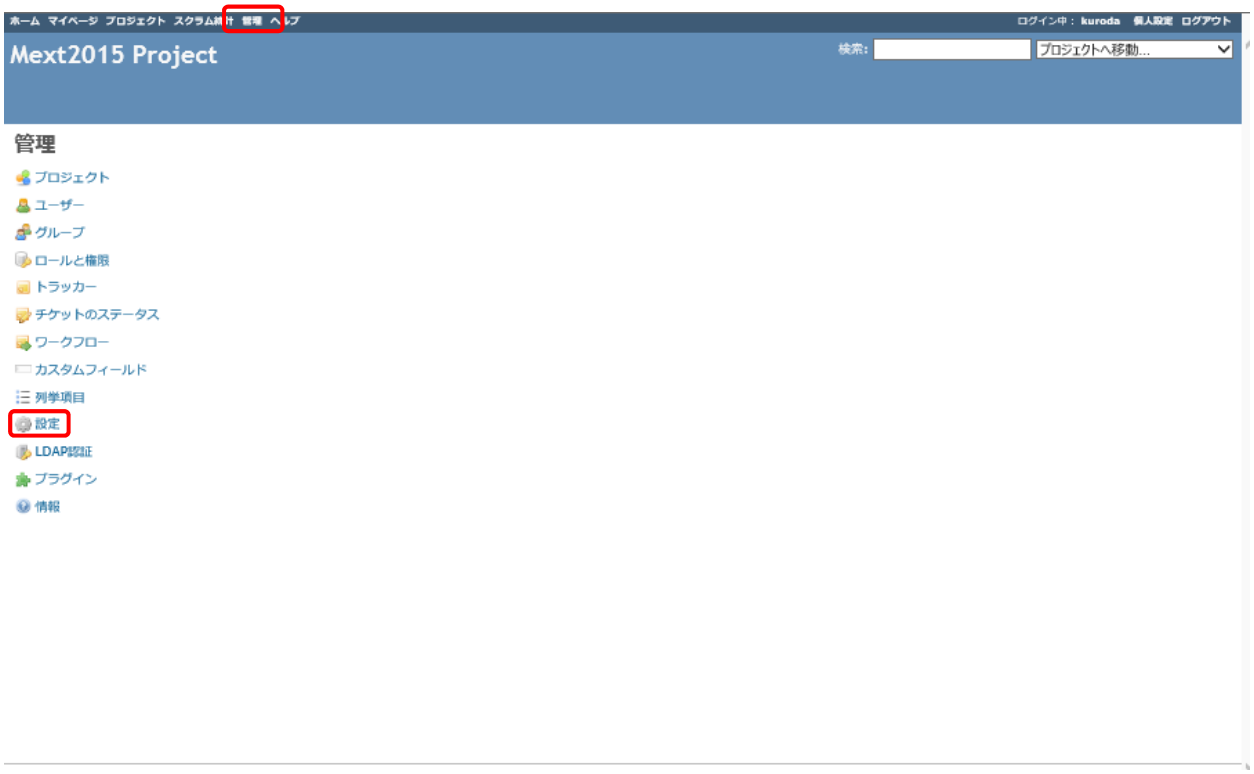
はじめに

RedmineはWebベースのプロジェクト管理システムです。
本資料では、 Redmine の環境設定方法について説明します。



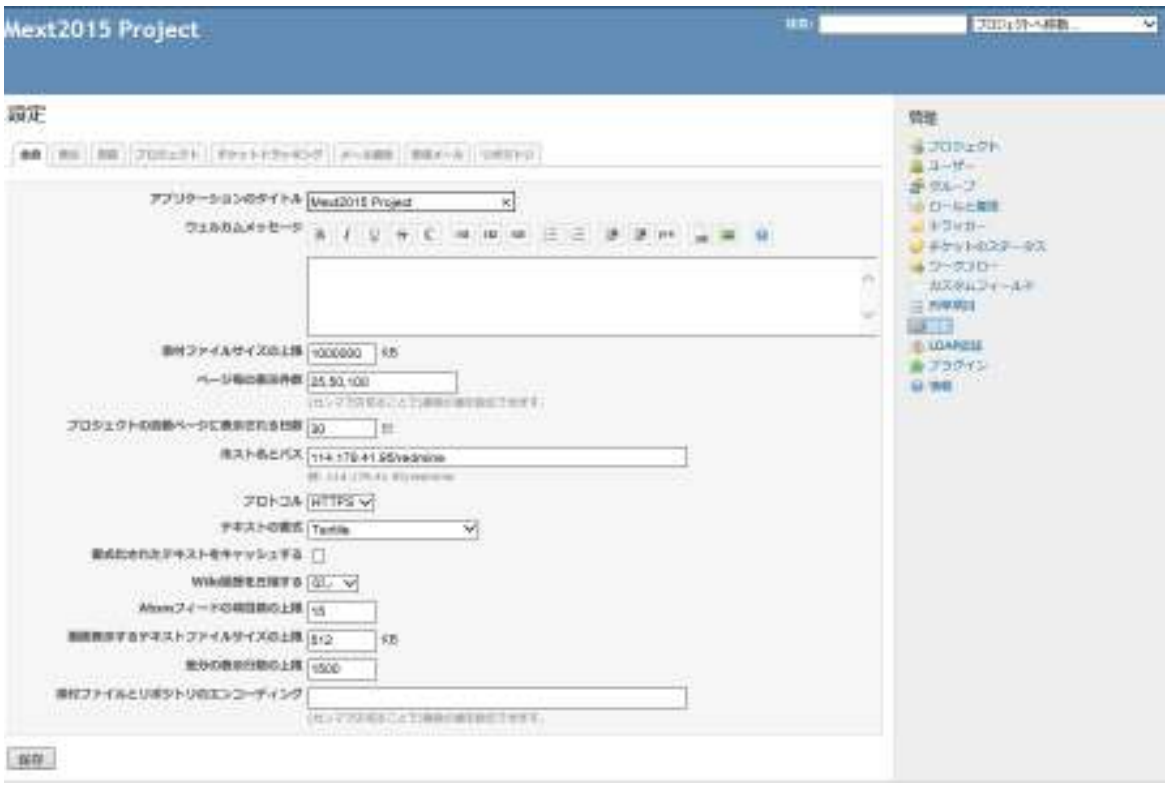
Redmineの設定

1.Redmineの設定



5

1.Redmineの設定（全般）

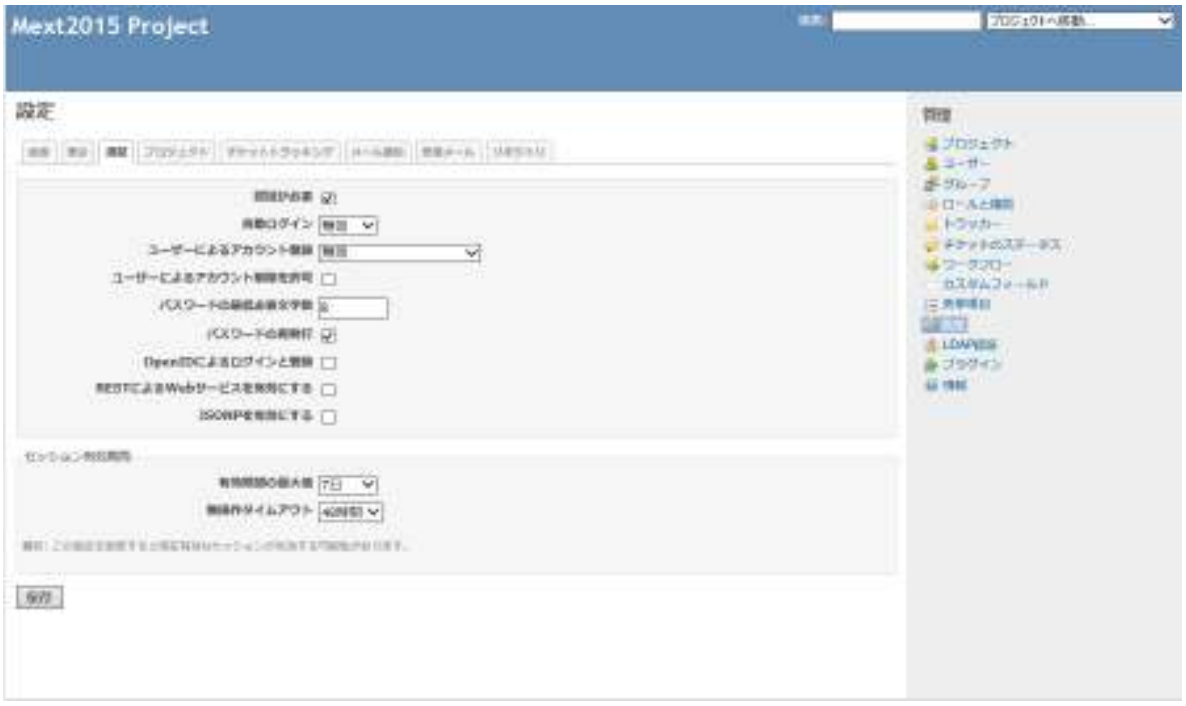


6

1.Redmineの設定（表示）

7

1.Redmineの設定（認証）

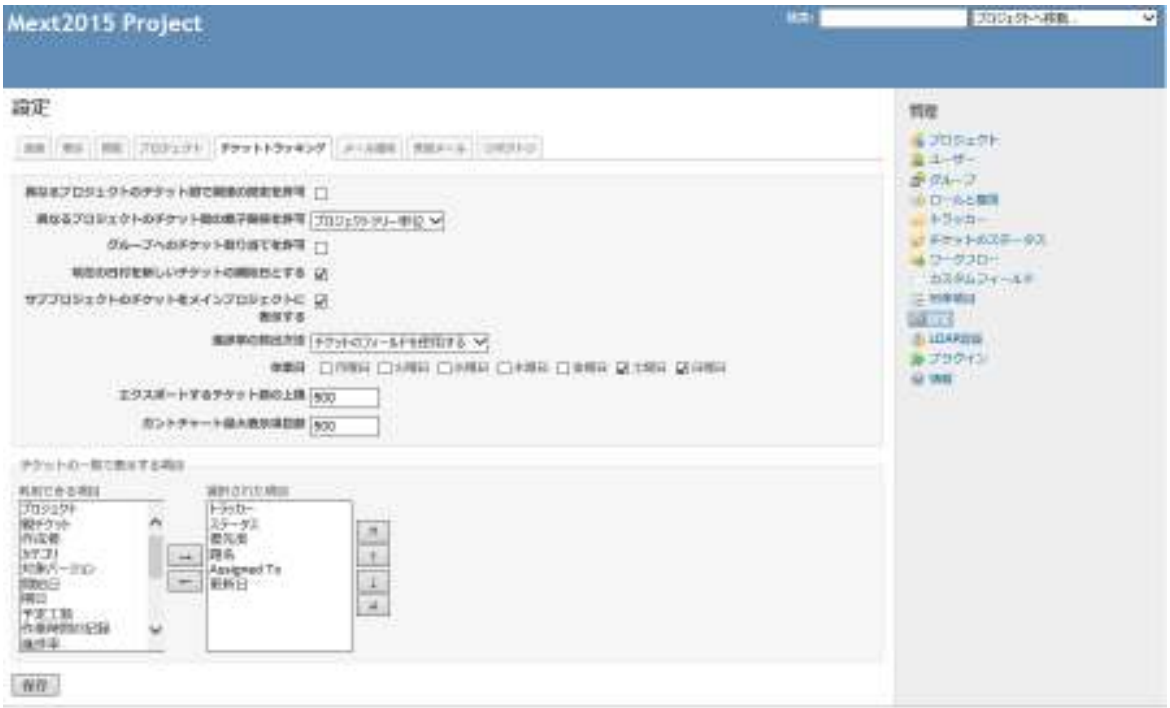


8

1.Redmineの設定（プロジェクト）

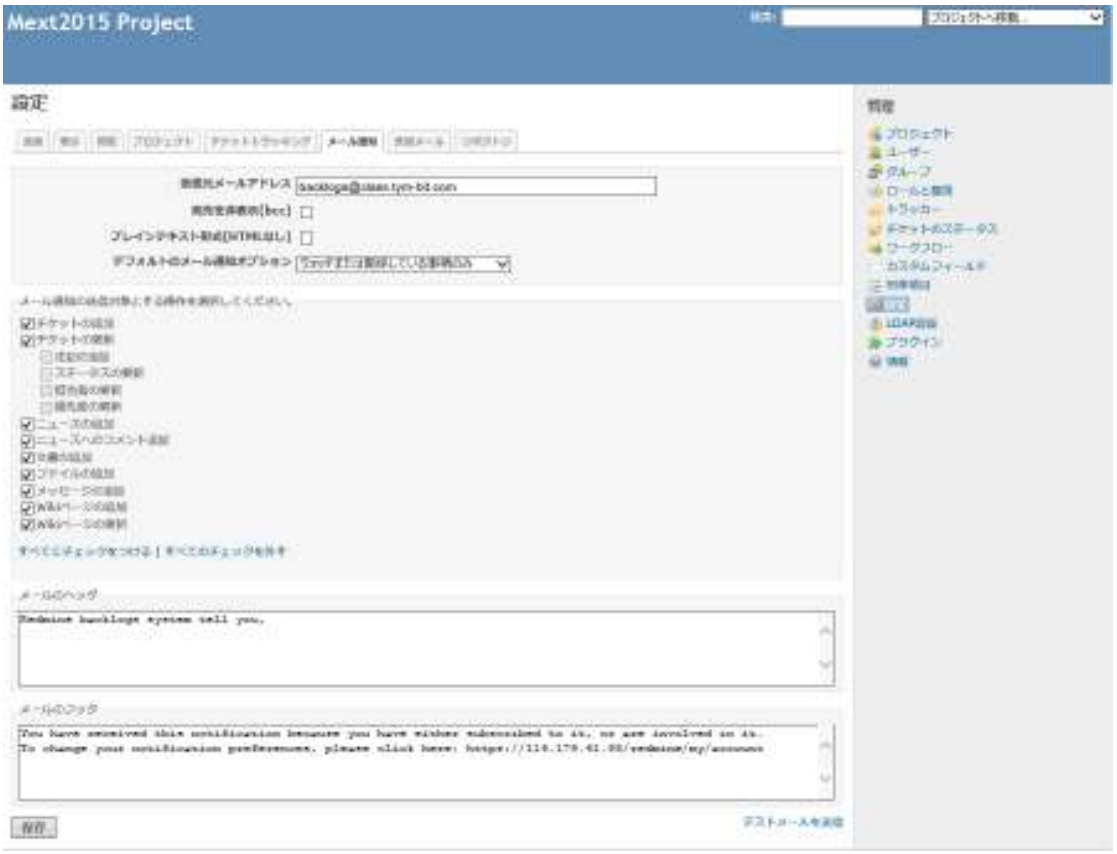
9

1.Redmineの設定（チケットトラッキング）



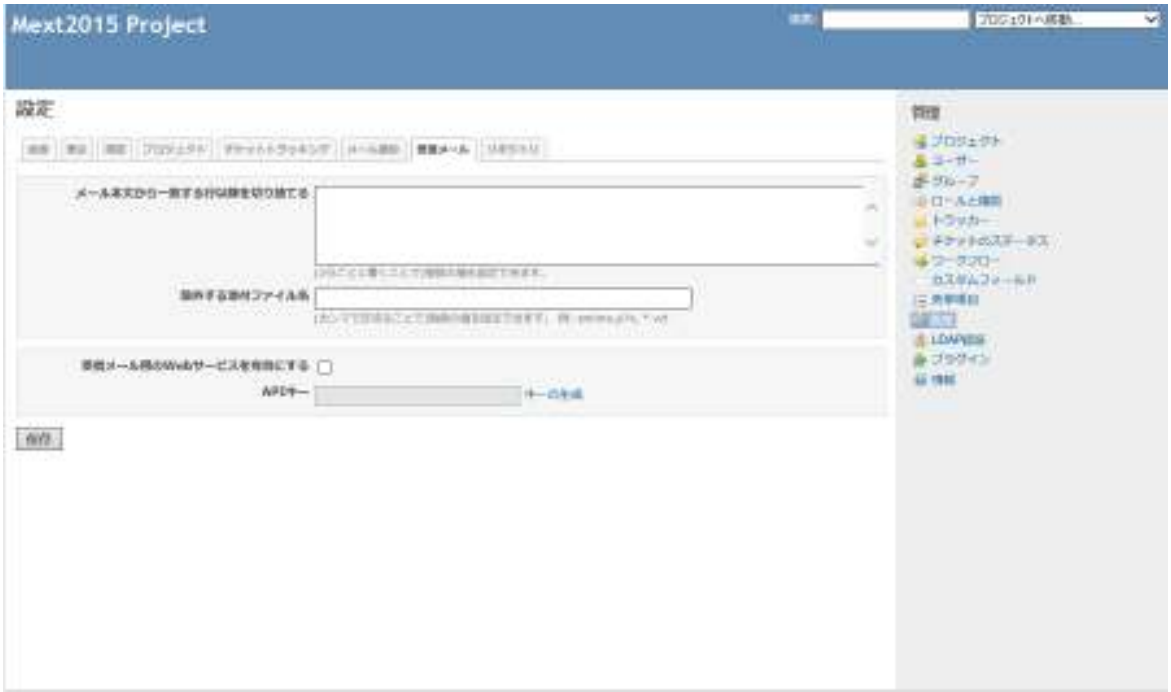
10

1.Redmineの設定（メール通知）



11

1.Redmineの設定（受信メール）



12

1.Redmineの設定（リポジトリ）

13

プロジェクトの作成

14

2.プロジェクトの作成（プロジェクト一覧）

以下のプロジェクトを作成します。

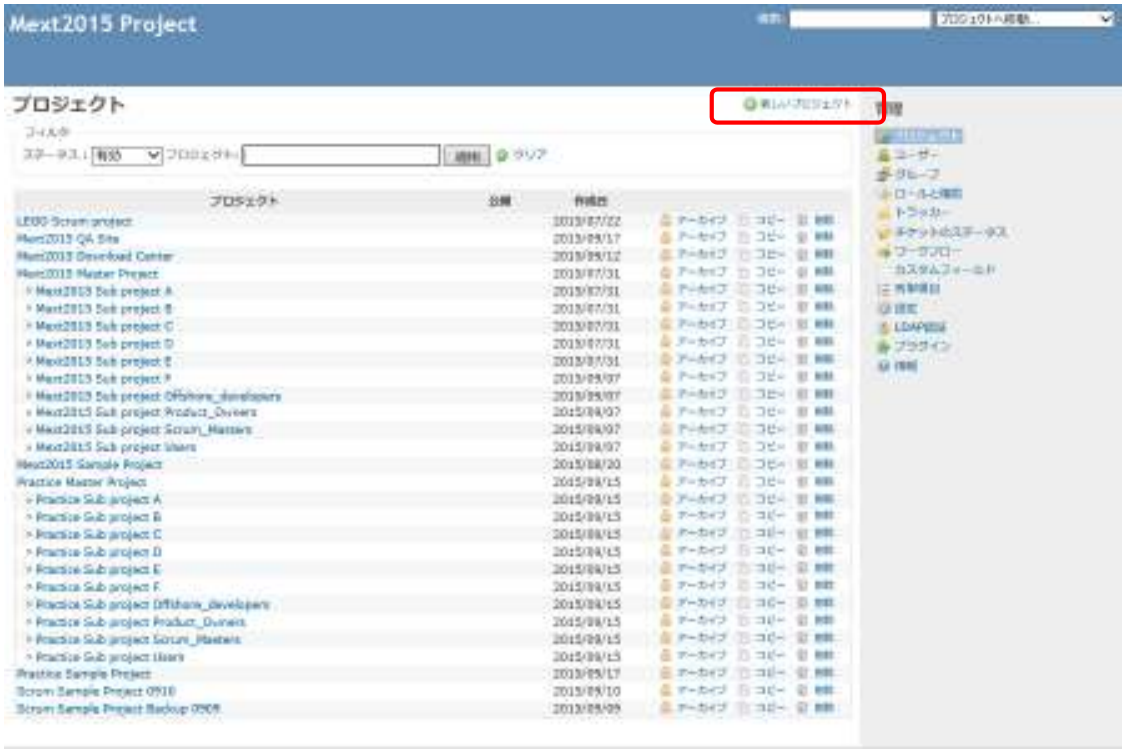
プロジェクト名	サブプロジェクト名	説明
Scrum Development. Master Project	Scrum Dev. Project Team A	スクラムチームAのプロジェクト
	Scrum Dev. Project Team B	スクラムチームBのプロジェクト
	Scrum Dev. Project Team C	スクラムチームCのプロジェクト
	Scrum Dev. Project Team D	スクラムチームDのプロジェクト
	Scrum Dev. Project Team E	スクラムチームEのプロジェクト
	Scrum Dev. Project PO	プロダクトオーナーのプロジェクト
	Scrum Dev. Project SM	スクラムマスターのプロジェクト
	Scrum Dev. Project Oversea	開発者（海外）のプロジェクト

15

2.プロジェクトの作成

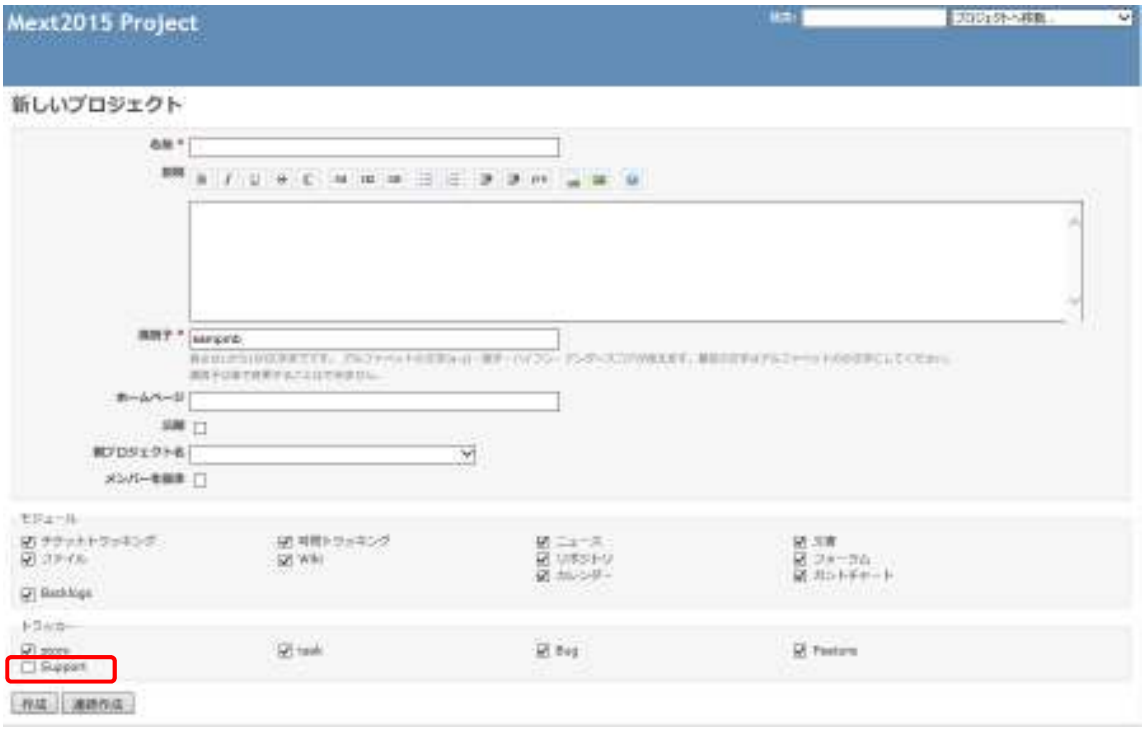
16

2.プロジェクトの作成(新しいプロジェクトの作成)



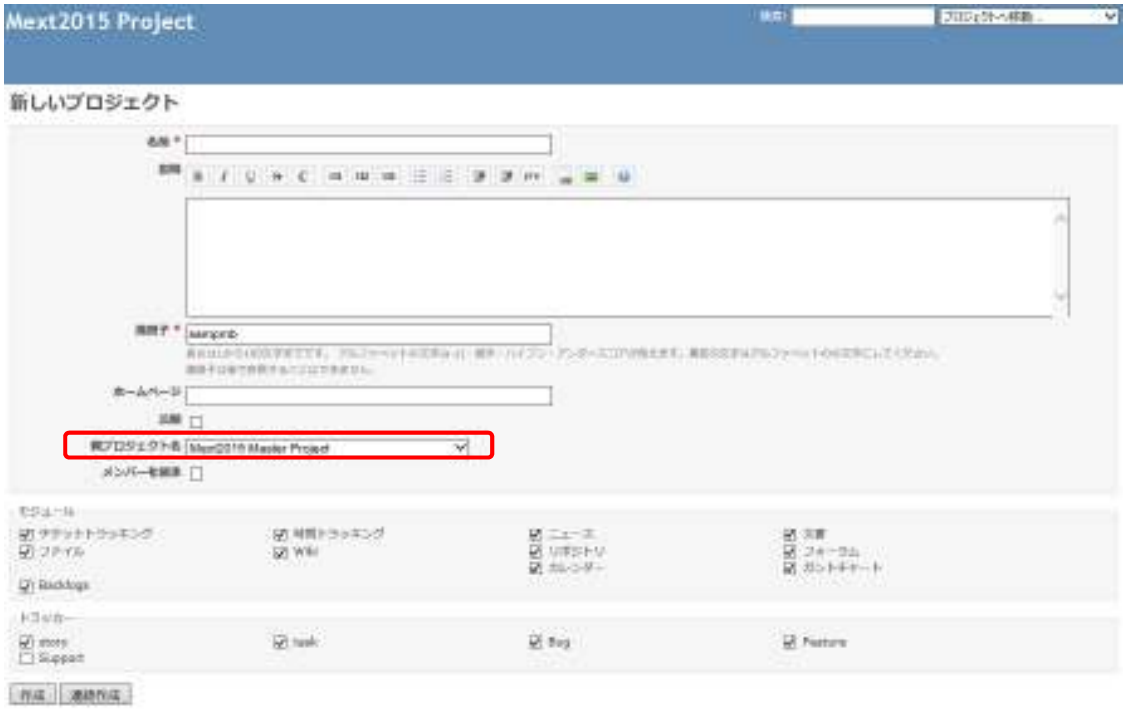
17

2.プロジェクトの作成（親プロジェクトの作成）



18

2.プロジェクトの作成（サブプロジェクトの作成）



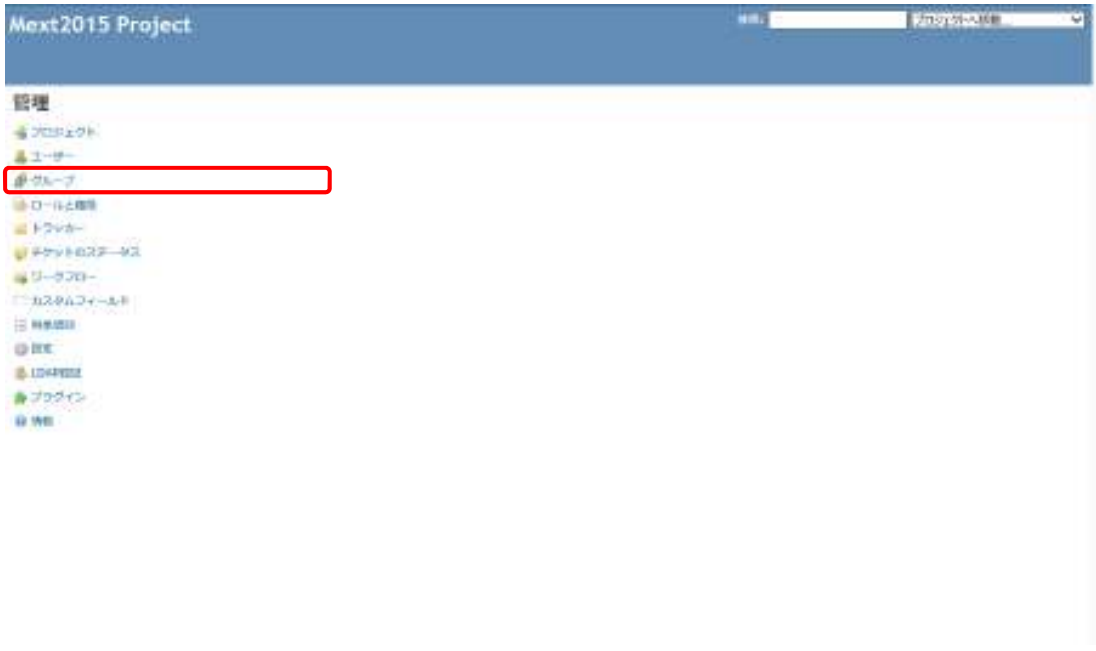
グループの作成

3.グループの作成（グループ一覧）

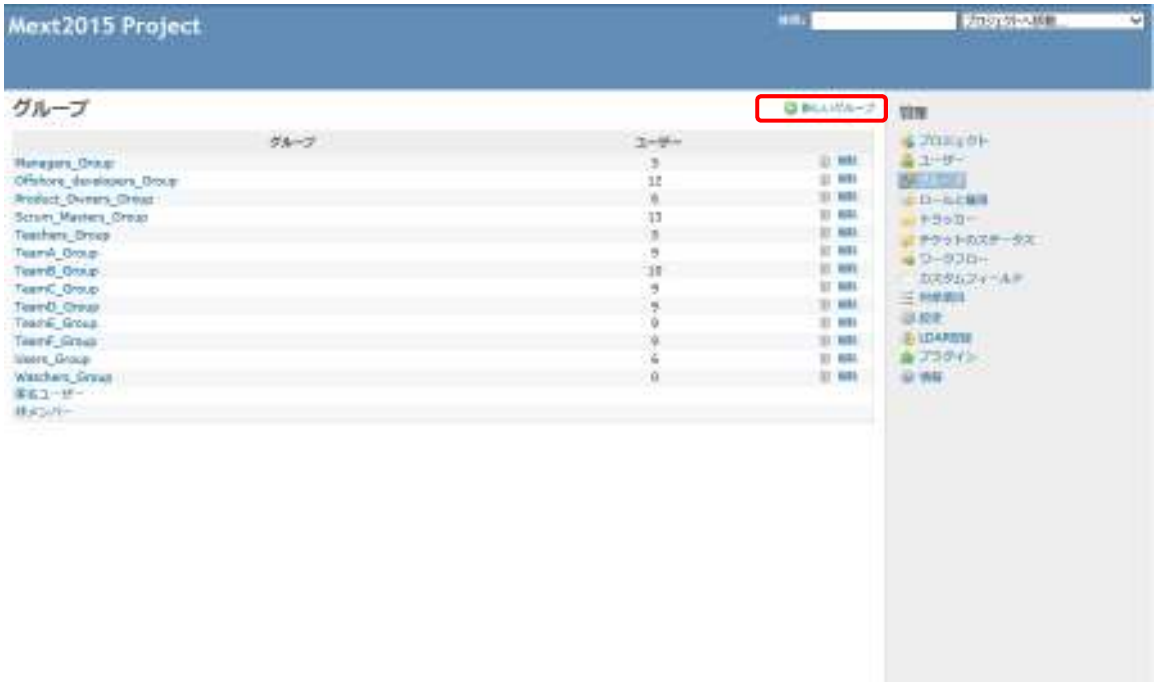
以下のグループを作成します。

グループ	説明
Managers_Group	マネージャのグループ
Product_Owners_Group	プロダクトオーナーのグループ
Scrum_Masters_Group	スクラムマスタのグループ
Teachers_Group	講師のグループ
TeamA_Group	スクラムチームＡのグループ
TeamB_Group	スクラムチームＢのグループ
TeamC_Group	スクラムチームＣのグループ
TeamD_Group	スクラムチームＤのグループ
TeamE_Group	スクラムチームEのグループ
Watchers_Group	閲覧者のグループ

3.グループの作成

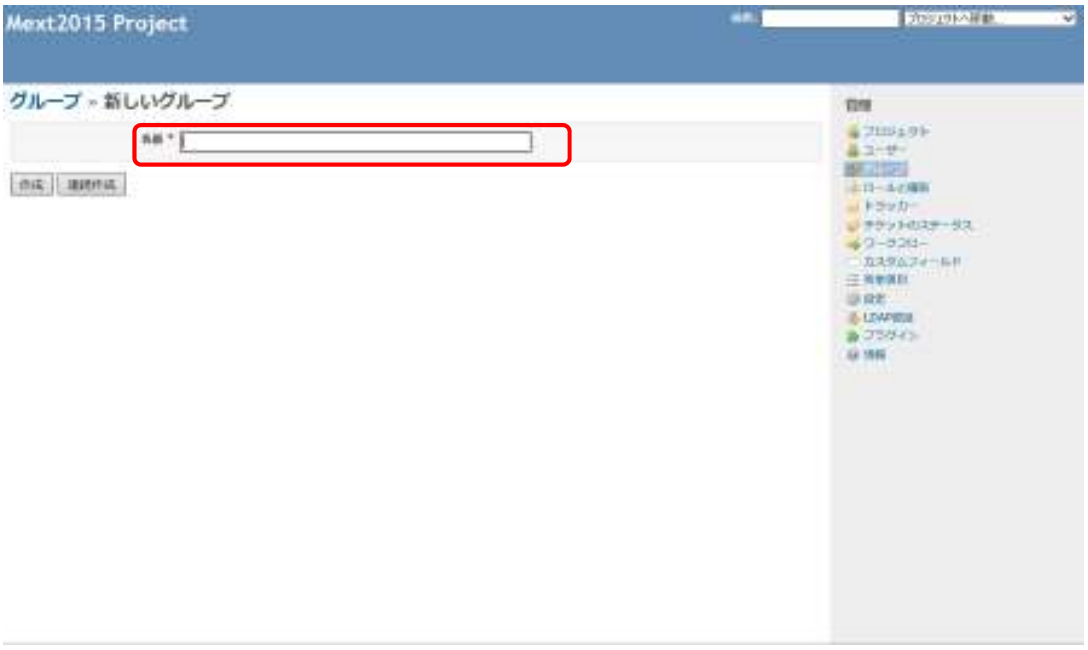


3.グループの作成（新しいグループの作成）



23

3.グループの作成（新しいグループの作成）



24

ロールと権限の作成

4.ロールと権限の作成（ロール一覧）

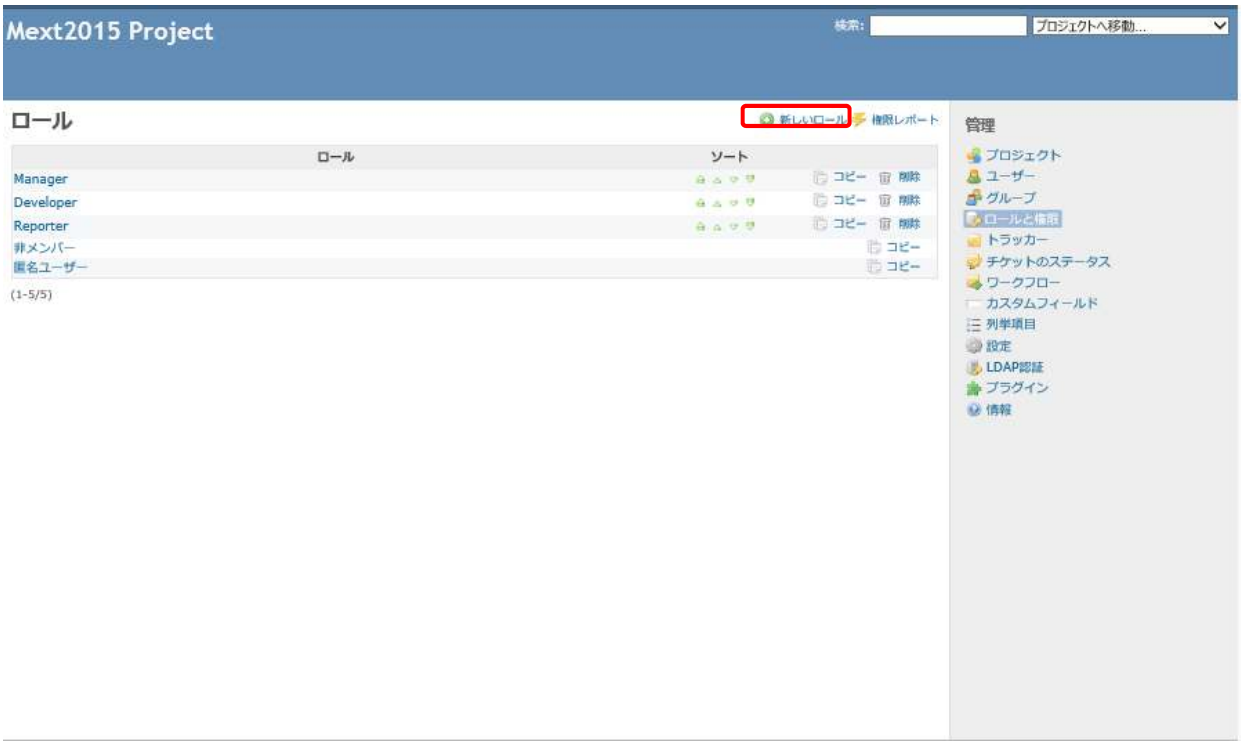
以下のロールを作成します。

ロール	説明
Manager	管理者の権限
Developer	開発者の権限
Reporter	報告者の権限

4.ロールと権限の作成

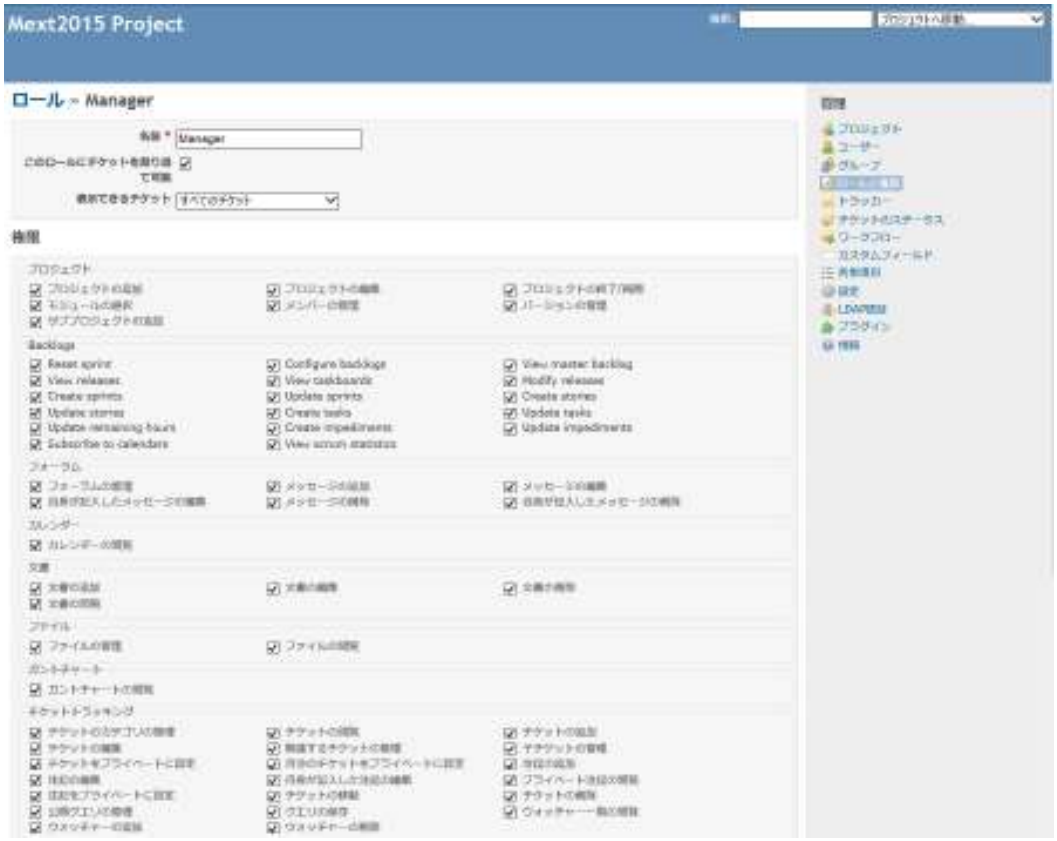
27

4.ロールと権限の作成（新しいロールの作成）



28

4.ロールと権限の作成（Manager権限）

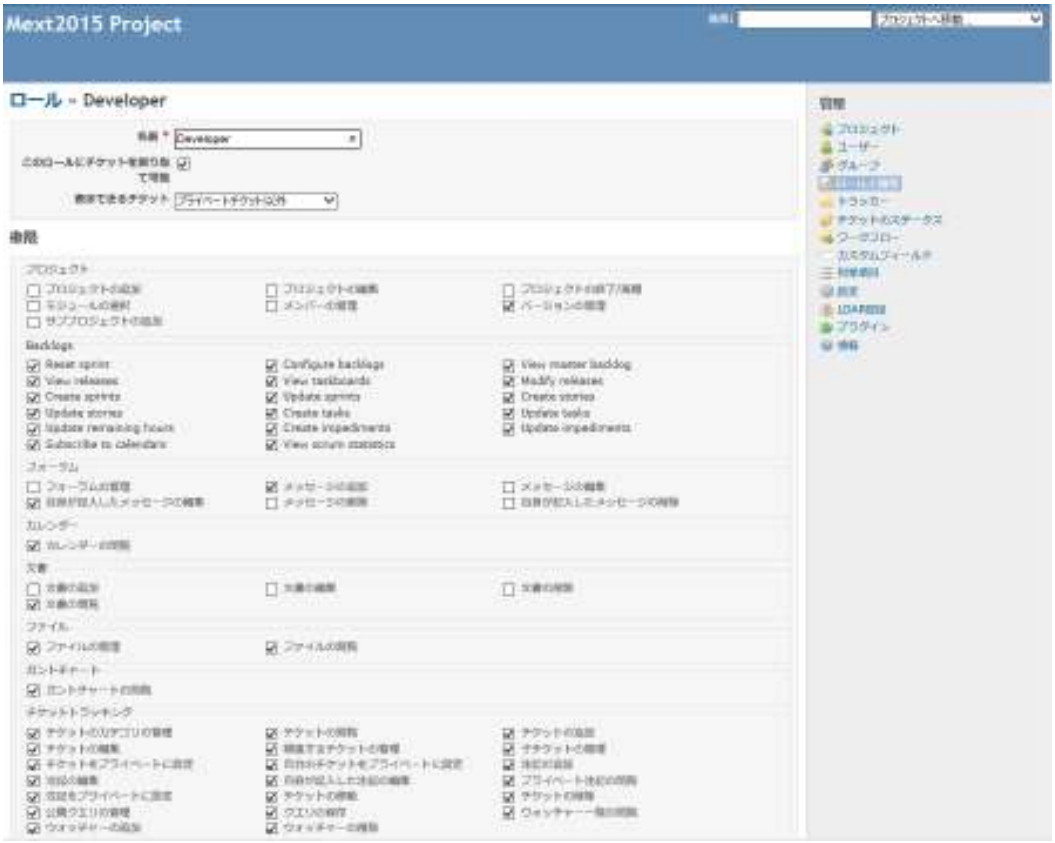


29

4.ロールと権限の作成（Manager権限）

30

4.ロールと権限の作成（Developer権限）



4.ロールと権限の作成（Developer権限）

ニュース
- [ ] ニュースの管理
- [x] ニュースへのコメント

リポジトリ
- [ ] リポジトリの管理
- [x] リポジトリの閲覧
- [x] 更新履歴の閲覧
- [x] コミット権限
- [x] 関連するチケットの管理

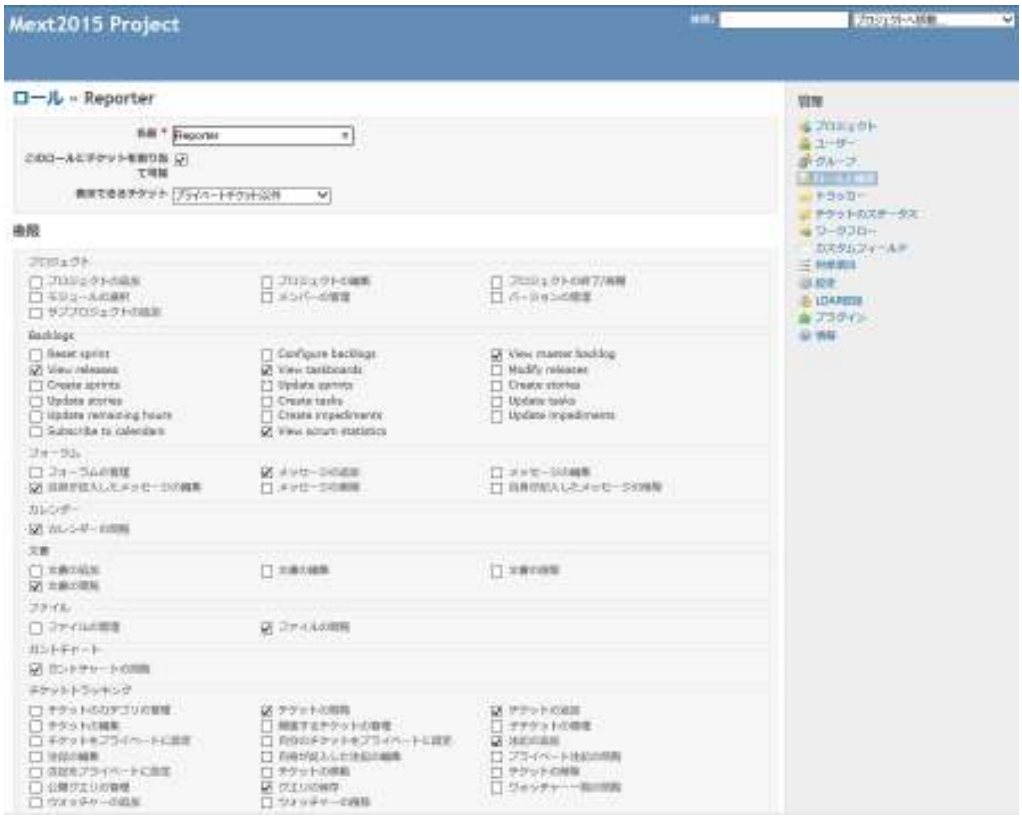
時間トラッキング
- [x] 作業時間の記入
- [x] 作業時間の閲覧
- [ ] 作業時間の編集
- [ ] 自身が記入した作業時間の編集
- [ ] 作業分類（時間トラッキング）の管理

Wiki
- [ ] Wikiの管理
- [ ] Wikiページ名の変更
- [x] Wikiページの削除
- [x] Wikiの閲覧
- [ ] Wikiページを他の形式にエクスポート
- [x] Wiki履歴の閲覧
- [x] Wikiページの編集
- [ ] 添付ファイルの削除
- [ ] Wikiページの凍結

すべてにチェックをつける | すべてのチェックを外す

保存

4.ロールと権限の作成（Reporter権限）



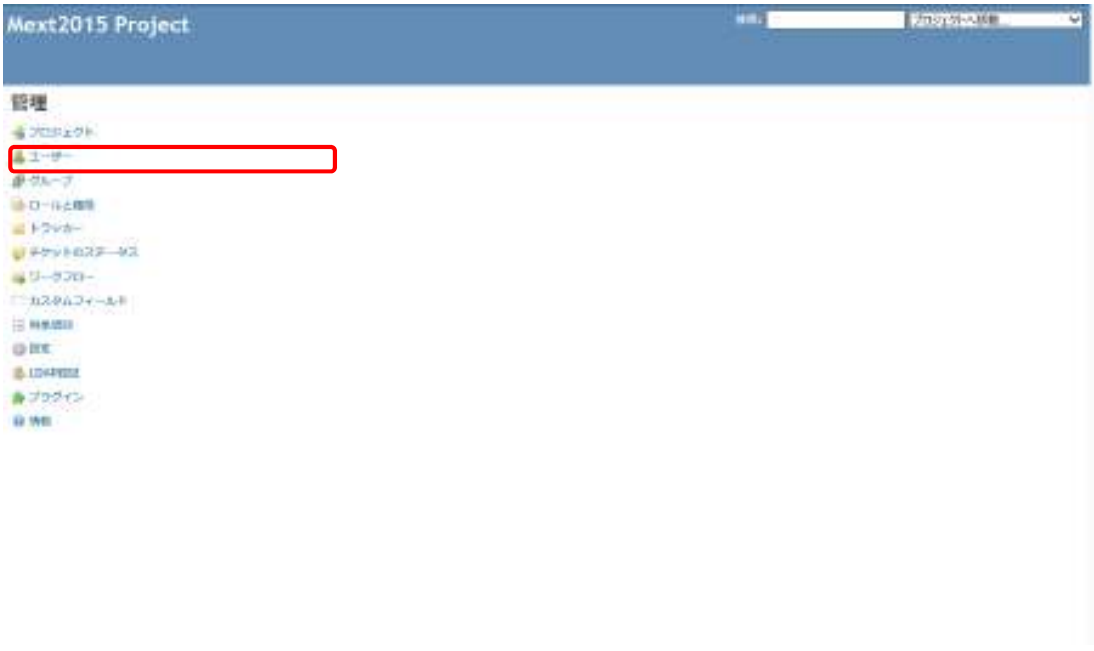
33

4.ロールと権限の作成（Reporter権限）

34

ユーザの作成

5.ユーザーの作成



5.ユーザーの作成（ユーザー一覧）

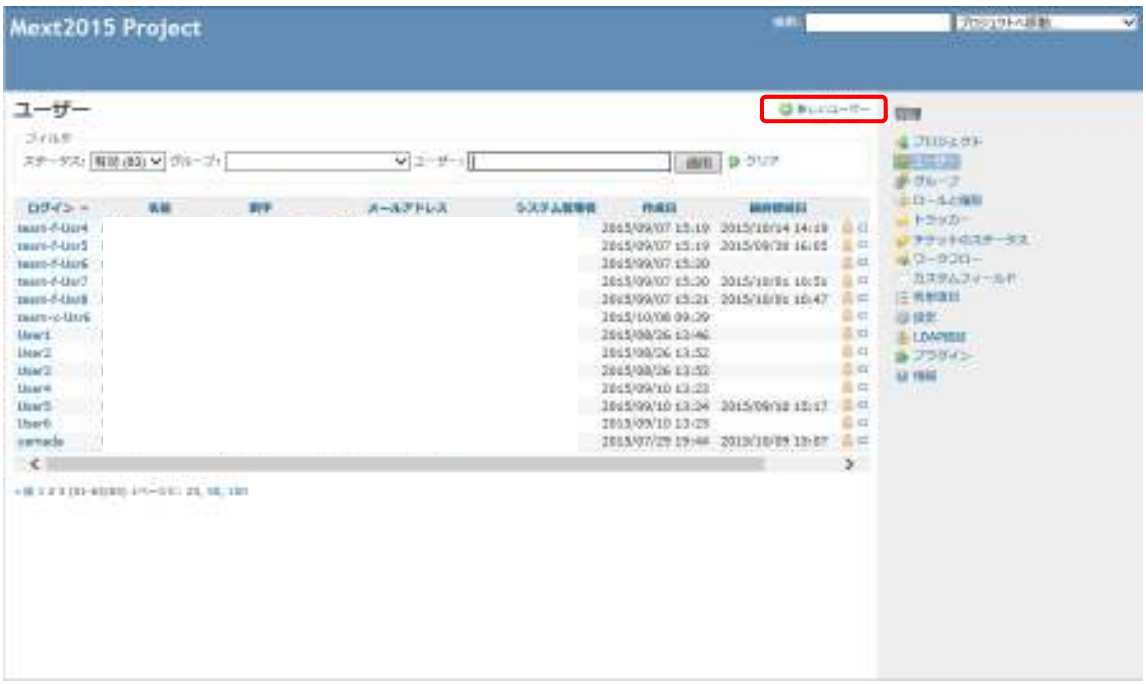
以下ユーザーを作成します。

チーム名	ユーザー	グループ									プロジェクト									ロール
		Managers_Group	Product_Owners_Group	Scrum_Masters_Group	Teachers_Group	TeamA_Group	TeamB_Group	TeamC_Group	TeamD_Group	TeamE_Group	Scrum Development. Master Project	Scrum Dev. Project Team A	Scrum Dev. Project Team B	Scrum Dev. Project Team C	Scrum Dev. Project Team D	Scrum Dev. Project Team E	Scrum Dev. Project PO	Scrum Dev. Project SM	Scrum Dev. Project Oversea	
講師チーム	ユーザー	○			○						○	○	○	○	○	○	○	○	○	Manager
スクラムチームA	プロダクトオーナー		○			○						○					○			Developer
	スクラムマスター			○		○						○						○		Developer
	開発者（日本側）					○						○								Developer
	開発者（日本側）					○						○								Developer
	開発者（日本側）					○						○								Developer
	開発者（海外側）					○						○							○	Developer
	開発者（海外側）					○						○							○	Developer
スクラムチームB	プロダクトオーナー		○				○						○				○			Developer
	スクラムマスター			○			○						○					○		Developer
	開発者（日本側）						○						○							Developer
	開発者（日本側）						○						○							Developer
	開発者（日本側）						○						○							Developer
	開発者（海外側）						○						○						○	Developer
	開発者（海外側）						○						○						○	Developer
スクラムチームC	プロダクトオーナー		○					○						○			○			Developer
	スクラムマスター			○				○						○				○		Developer
	開発者（日本側）							○						○						Developer
	開発者（日本側）							○						○						Developer
	開発者（日本側）							○						○						Developer
	開発者（海外側）							○						○					○	Developer
	開発者（海外側）							○						○					○	Developer

5.ユーザーの作成（ユーザー一覧）

チーム名	ユーザー	グループ									プロジェクト									ロール
		Managers_Group	Product_Owners_Group	Scrum_Masters_Group	Teachers_Group	TeamA_Group	TeamB_Group	TeamC_Group	TeamD_Group	TeamE_Group	Scrum Development. Master Project	Scrum Dev. Project Team A	Scrum Dev. Project Team B	Scrum Dev. Project Team C	Scrum Dev. Project Team D	Scrum Dev. Project Team E	Scrum Dev. Project PO	Scrum Dev. Project SM	Scrum Dev. Project Oversea	
スクラムチームD	プロダクトオーナー		○						○						○		○			Developer
	スクラムマスター			○					○						○			○		Developer
	開発者（日本側）								○						○					Developer
	開発者（日本側）								○						○					Developer
	開発者（日本側）								○						○					Developer
	開発者（海外側）								○						○				○	Developer
	開発者（海外側）								○						○				○	Developer
スクラムチームE	プロダクトオーナー		○							○						○	○			Developer
	スクラムマスター			○						○						○		○		Developer
	開発者（日本側）									○						○				Developer
	開発者（日本側）									○						○				Developer
	開発者（日本側）									○						○				Developer
	開発者（海外側）									○						○			○	Developer
	開発者（海外側）									○						○			○	Developer

5.ユーザーの作成（新しいユーザーの作成）

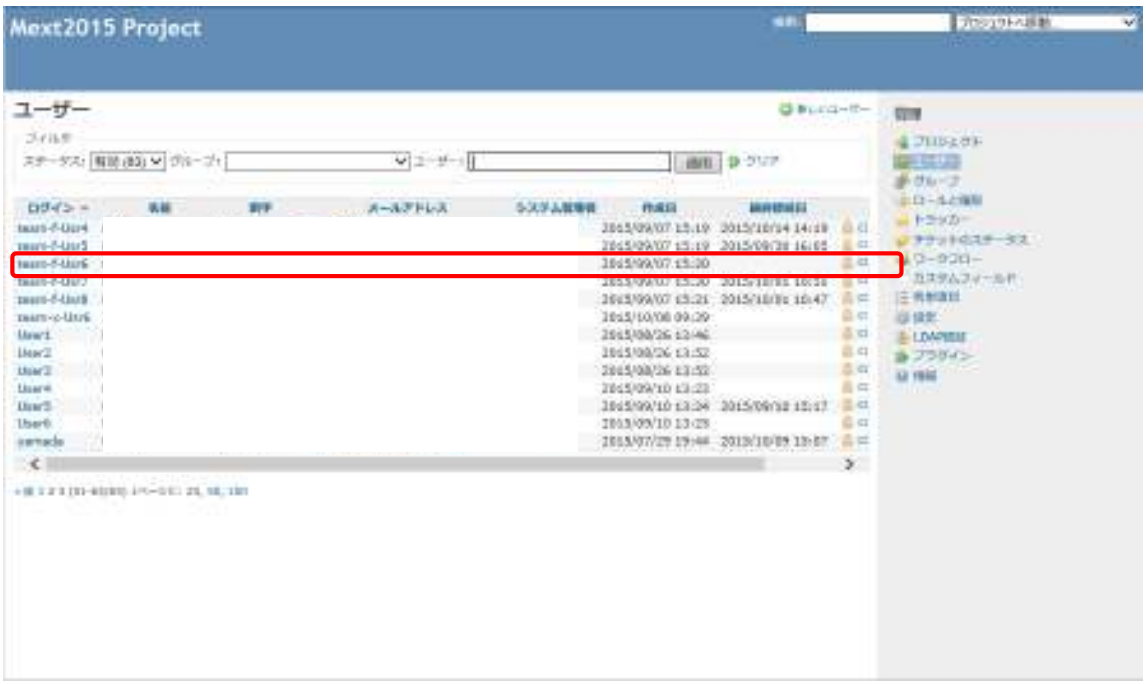


39

5.ユーザーの作成（新しいユーザーの作成）

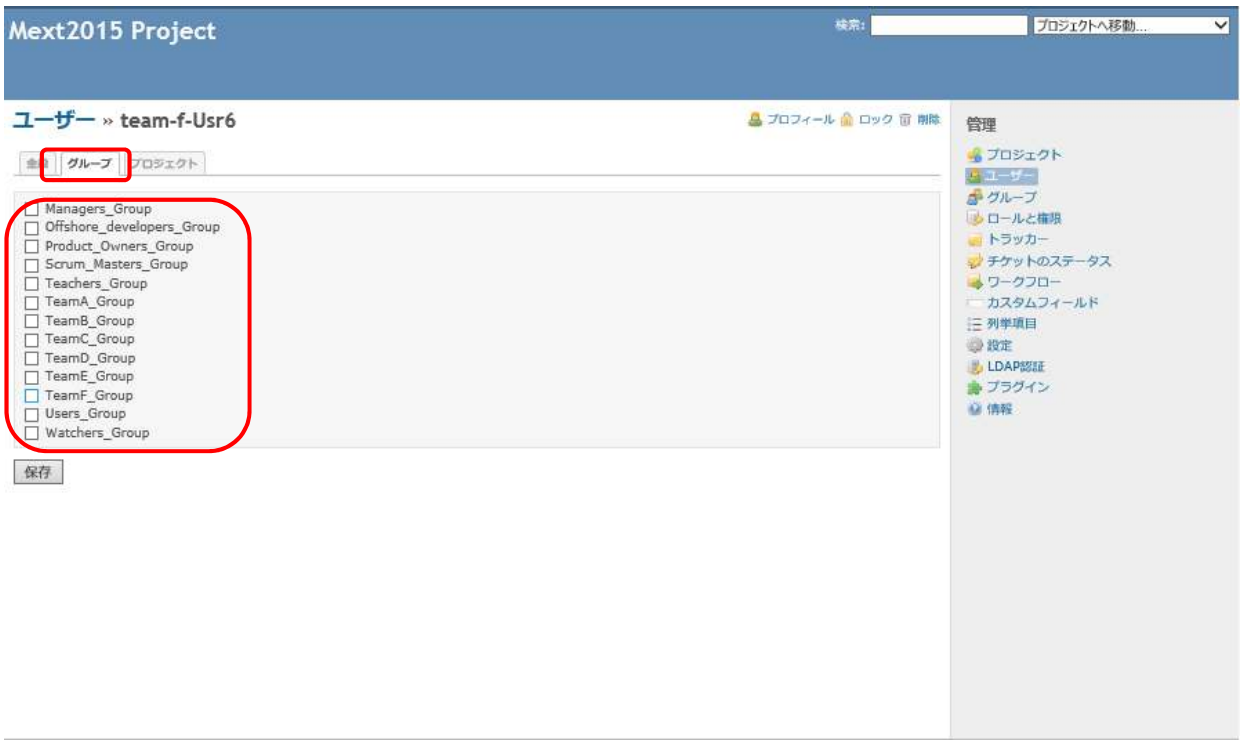
40

5.ユーザーの作成（ユーザーの選択）



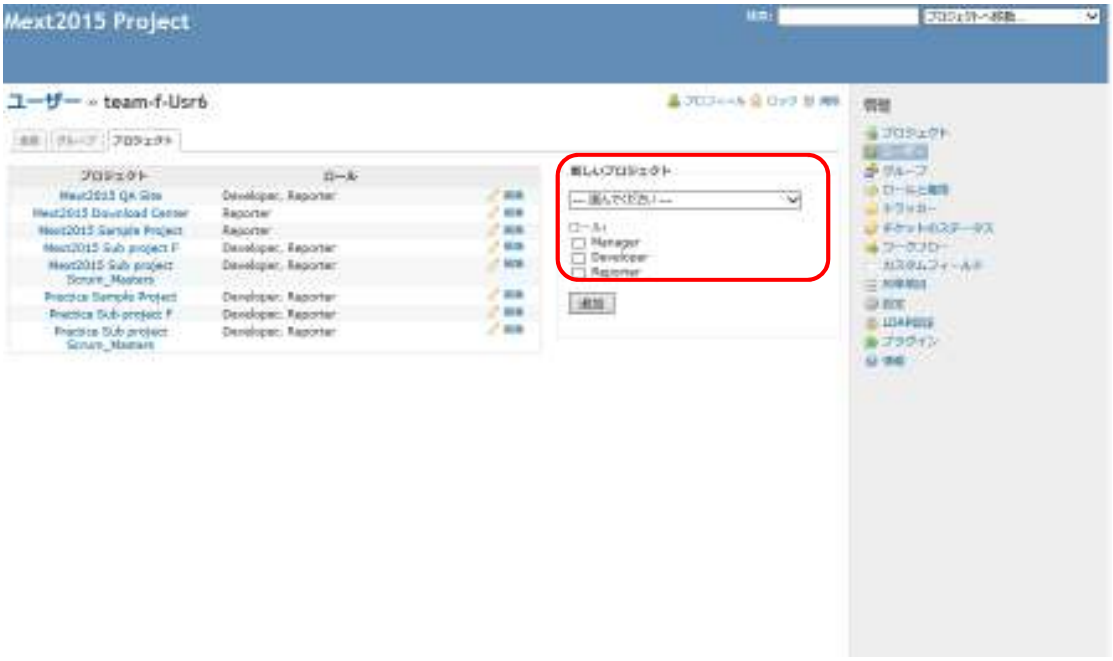
41

5.ユーザーの作成（グループの設定）



42

5.ユーザーの作成（プロジェクトの設定）



トラッカーの作成

6.トラッカーの作成（トラッカー一覧）

以下のトラッカーを作成します。

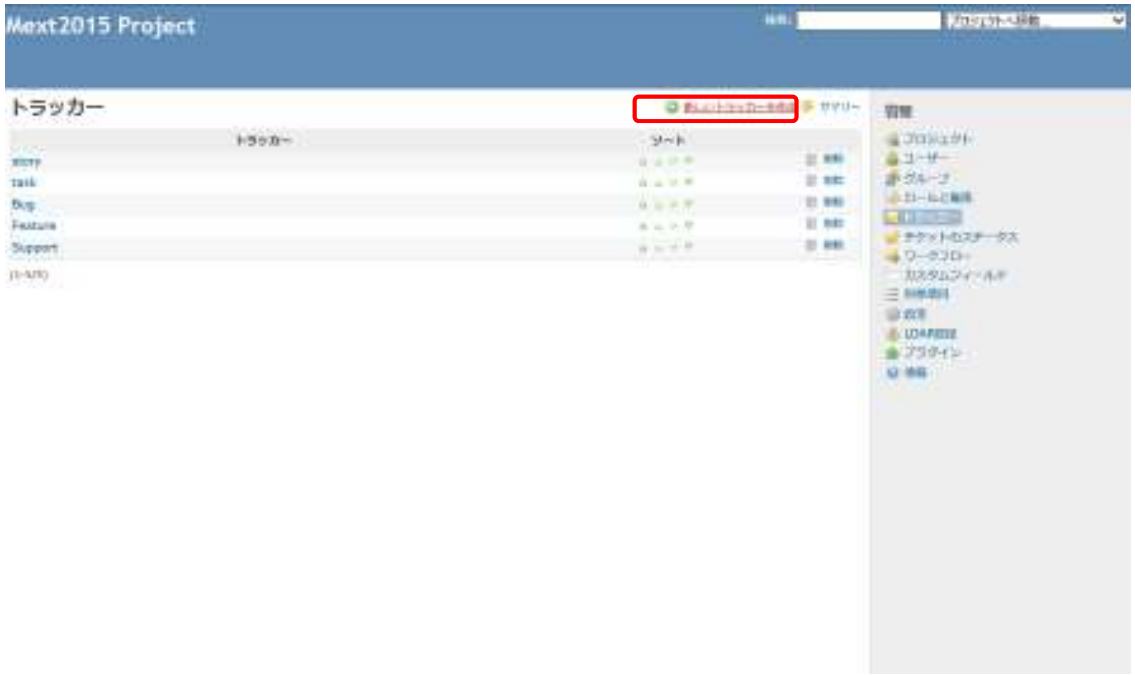
プロジェクト名	トラッカー	説明
Mext2015 Master Project	story	プロダクトバックログのトラッカー
	task	スプリントバックログのトラッカー

45

6.トラッカーの作成

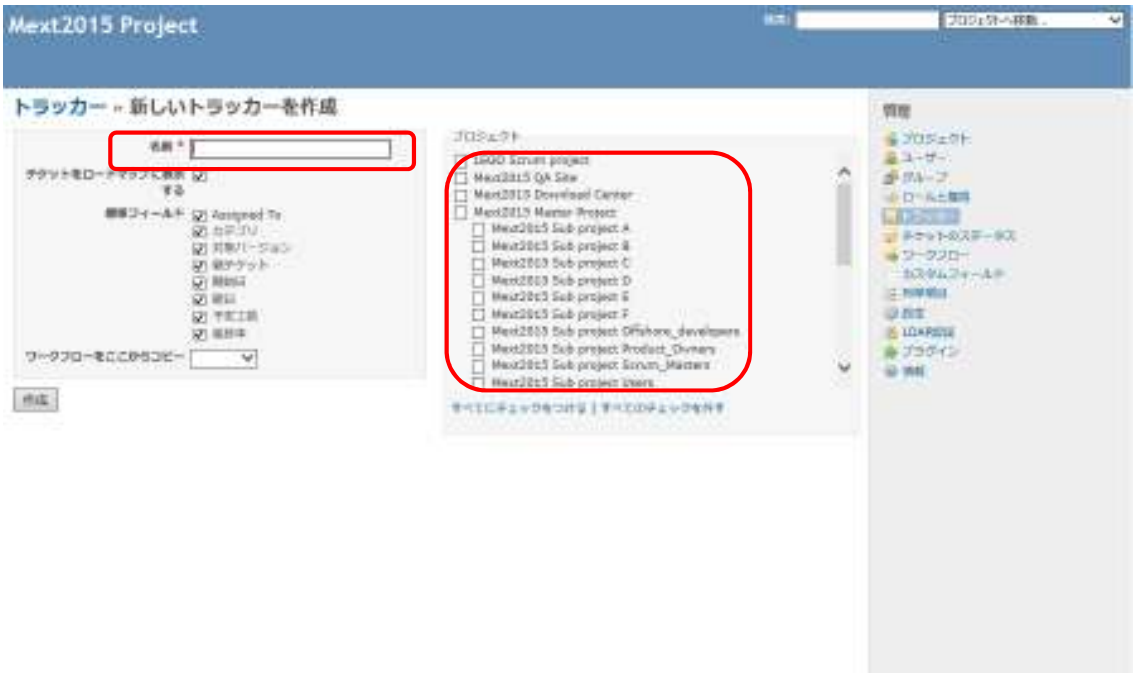
46

6.トラッカーの作成（新しいトラッカーの作成）



47

6.トラッカーの作成（プロジェクトを選択）



48

チケットのステータス作成

7.チケットのステータス作成（チケットのステータス一覧）

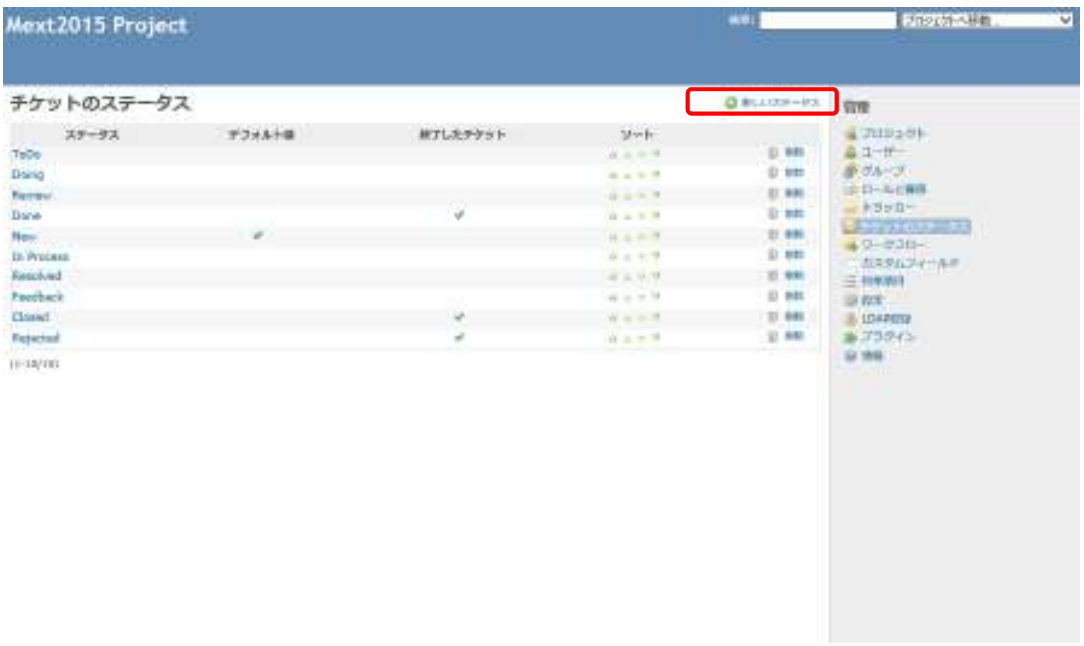
以下のチケットのステータスを作成します。

プロジェクト名	チケットのステータス	説明
Mext2015 Master Project	ToDo	作業前（やること）のチケット
	Doing	進行中（仕掛中）のチケット
	Review	レビューするチケット
	Done	完了したチケット

7.チケットのステータス作成

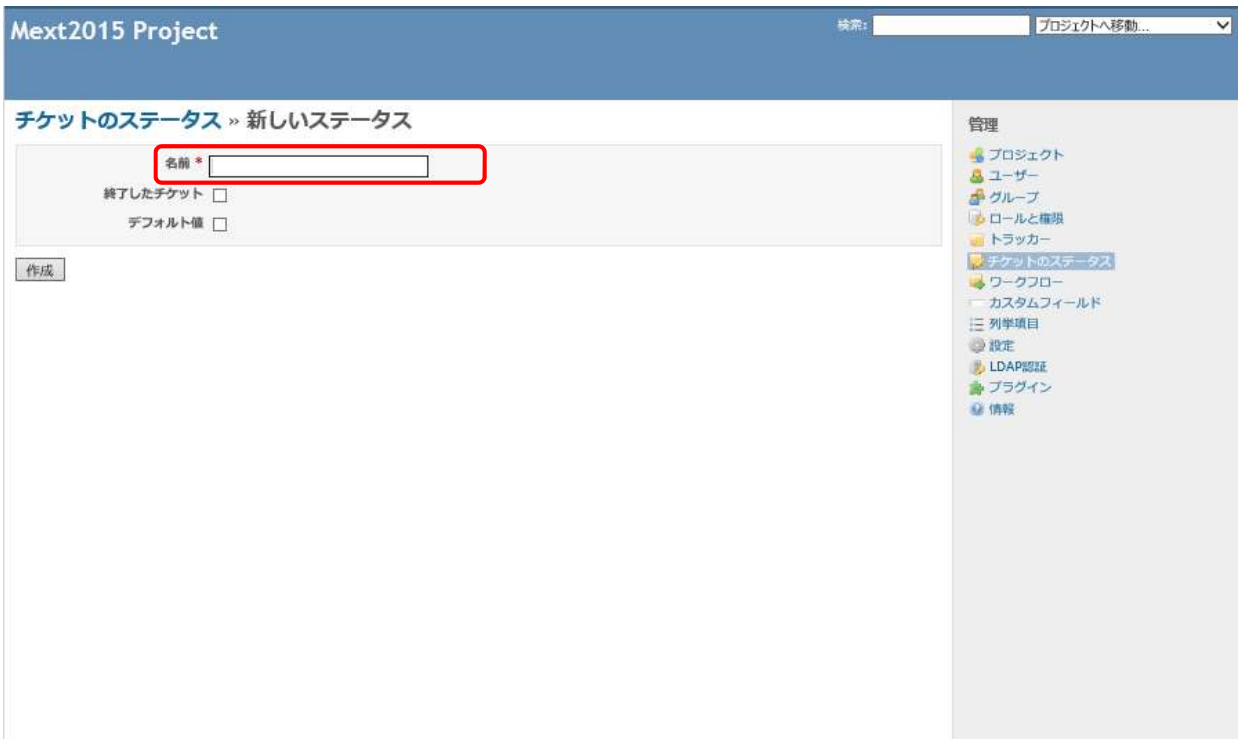
51

7.チケットのステータス作成（新しいステータス作成）



52

7.チケットのステータス作成（新しいステータス作成）

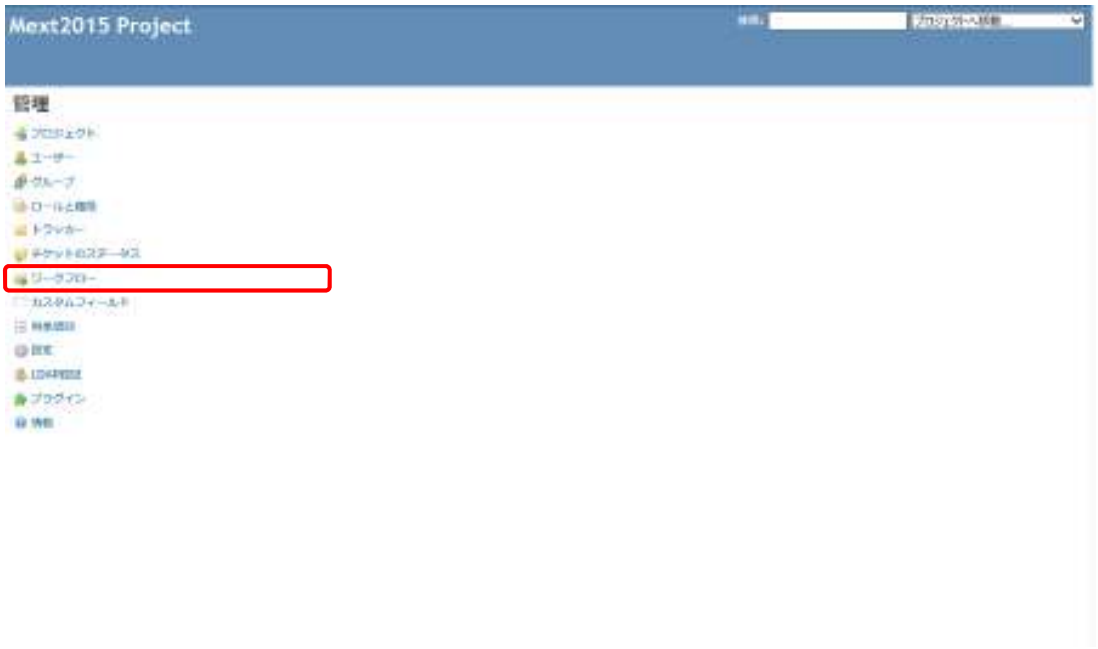


53

ワークフローの作成

54

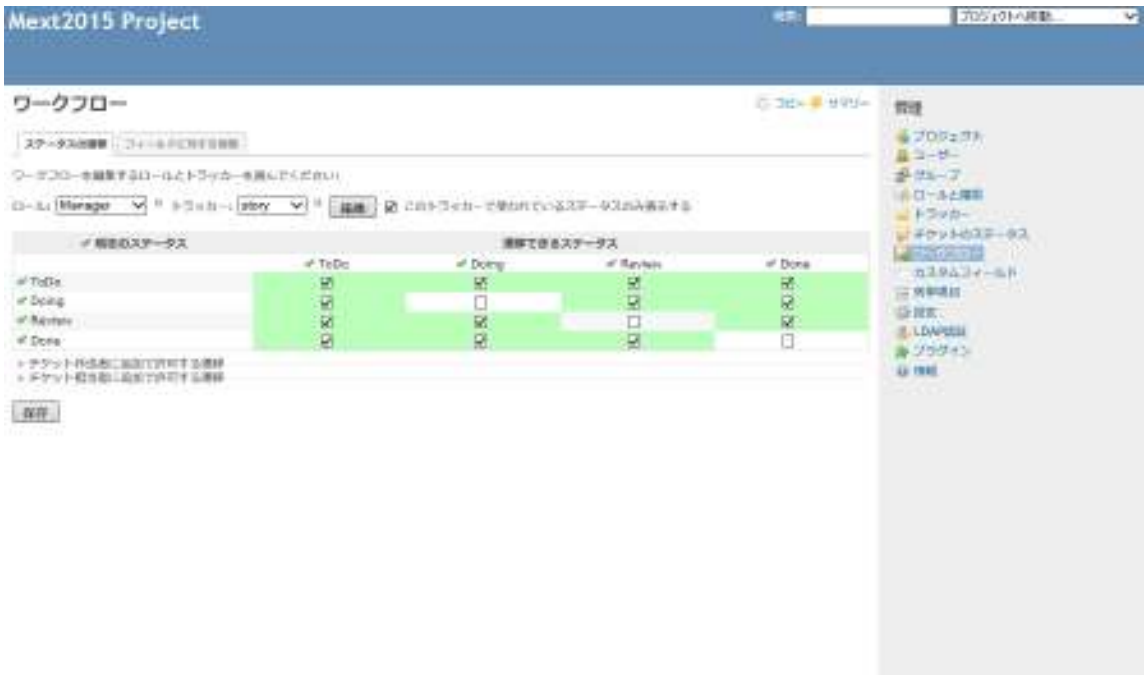
8.ワークフローの作成



55

8.ワークフローの作成（ステータスの遷移）

ロール　　：Manager , Developer , Reporter
トラッカー　：story , task

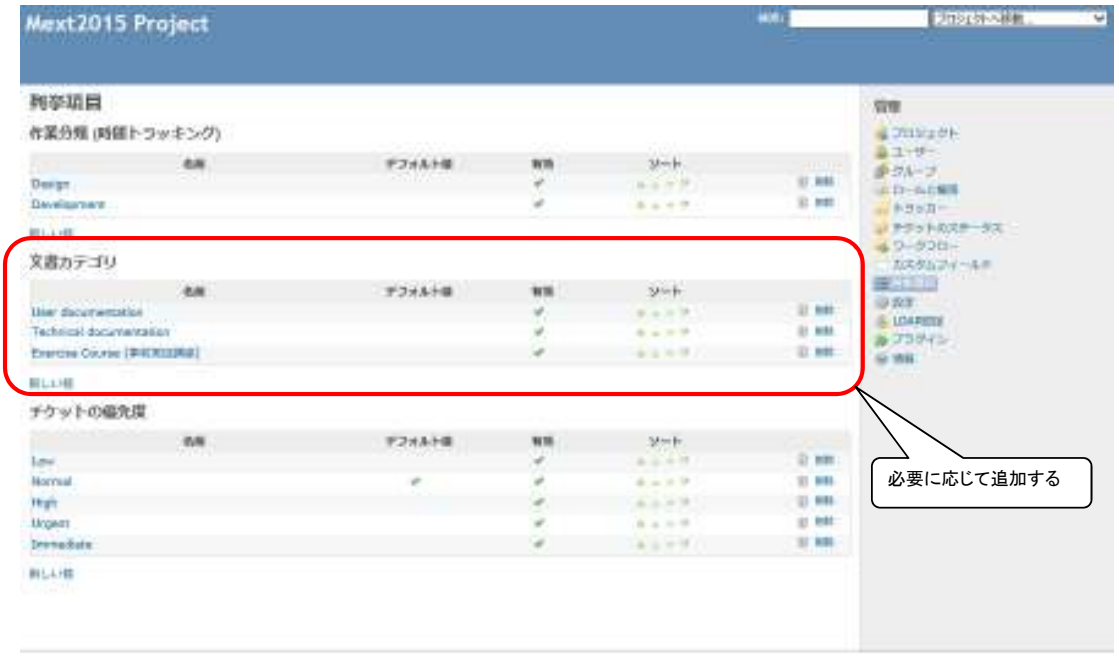


56

列挙項目の作成

9.列挙項目の作成

9.列挙項目の作成

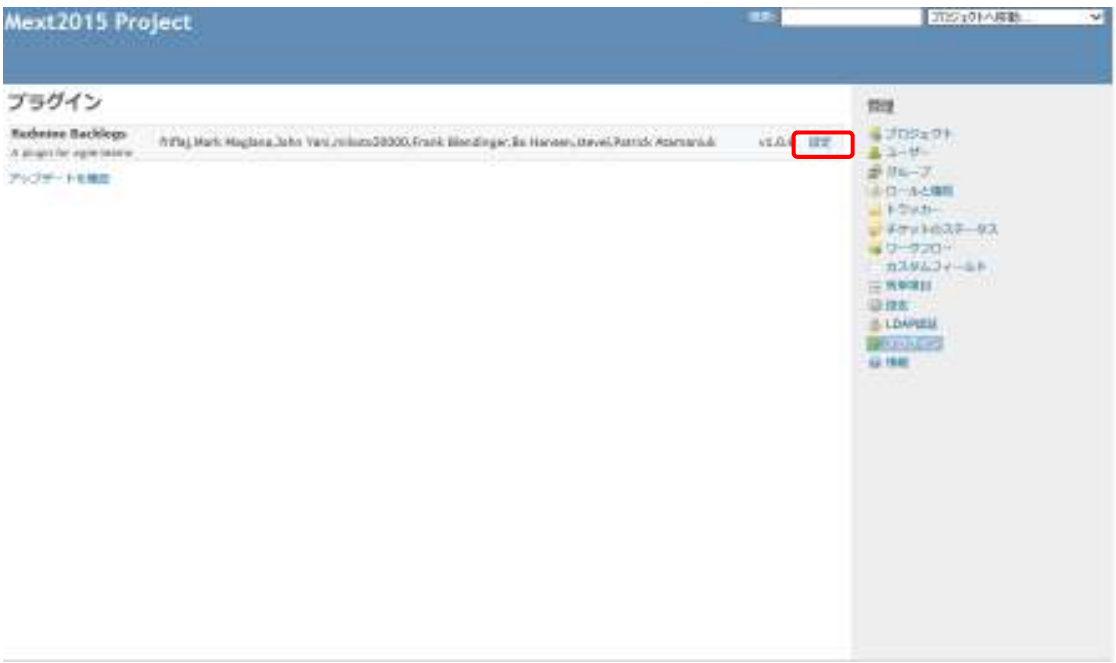


プラグインの設定

10.プラグインの設定

61

10.プラグインの設定



62

10.プラグインの設定

10.プラグインの設定

地方型グローバルITエンジニアの育成
ＰＢＬカリキュラム

教員育成指導書

２０１５年度版

富山情報ビジネス専門学校編

目次

１．はじめに.....1
２．目的.....1
２．１．当カリキュラムの目標.....1
２．２．概要.....1
２．３．前提知識.....1
３．構成.....2
３．１．カリキュラム全体の構成.....2
３．２．事前準備講座 e-Learning.....2
３．３．カリキュラム説明会及び、開発手法の理解、受講環境の構築と各種ツール操作の習得、ユーザー要件の説明、スクラム入門（第１週～第２週：全２週間）.....2
３．４．スプリント１の演習（第３週～第６週：全４週間）.....3
３．５．スプリント２の演習（第７週～第１０週：全４週間）.....3
３．６．スプリント３の演習（第１１週～第１４週：全４週間）.....4
３．７．成果発表及び、最終評価（第１５週）.....4
４．おわりに.....4

１．はじめに

当資料は、「地方型グローバル IT エンジニアの育成ＰＢＬカリキュラム」を学生に教授する講師を育成する為の補助教材として使用します。対象としているカリキュラムは、スクラム開発手法をはじめ、英語によるコミュニケーション、Java 言語、Web サーバ、データベース、PHP 言語、Redmine サーバ、クライアント開発環境(Android Studio)など多くの技術を駆使して実施する事を求められます。これらの技術をできるだけ早期に習得し、学生に教授できる様に学習指導要領と併用して当資料を利用して下さい。

２．目的

２．１．当カリキュラムの目標

グローバル IT エンジニアの育成を目指す為のカリキュラムです。設計作業からオフショア会社側担当者と共同で密接にコミュニケーションを取りながら開発を進めていく事ができる人材を育成することを目標とします。加えて海外エンジニアと協働でシステム開発の設計～プログラム開発～納品までの作業を行うことができるシステムエンジニアを指します。 グローバル IT エンジニア育成として、英語による読み・書きを中心とした海外エンジニアとの協働開発を体験します。

上記の目標を受講生が体験し、理解する為には、これらの前提となっている各種技術を講師側がしっかりと理解し、説明できる事が重要です。

２．２．概要

学習指導要領 １．２．概要に示されているように日本国内の顧客に対するシステム開発を国内エンジニアと海外エンジニアが協働で開発を進めます。スクラム開発技法のセオリーからすると両エンジニアが同じ開発場所で一緒に作業する事が最も効率が良いとされています。可能であれば日本人の学生と海外の学生(英語圏の協力者)が同じ場所で受講することをお薦めします。もし不可能であれば分散開発による作業となり、Skype などによる TV 会議システムやチャットシステム及び、e-mail 等を駆使して共同開発を実施します。尚、当カリキュラムは、分散開発作業で実施する場合を想定して構成されています。

２．３．前提知識

当カリキュラムを指導する講師は、以下の技術に関する知識を既得しているか、講義開始前までに習得しておく必要があります。上位の技術ほどより必要な前提知識となります。

- 英会話の基礎知識（簡単な日常会話程度で可）
- Java 言語によるアプリケーション開発の経験
- アンドロイド アプリケーションの開発経験
- PHP 言語による開発知識
- MySQL データベースサーバーの構築及び設定知識
- Redmine サーバの構築と操作に関する知識
- Web サーバの構築及び設定経験

・　チームによるシステム開発の経験(ウォータフォール手法でも可)
・　Android Studio（アンドロイドアプリケーション開発環境）での開発方法の知識
・　分散開発となる場合は、Skype 等による TV 会議システム及びチャットシステムの運用知識

※※ 上記前提知識で授業開始までに不足するものがあった場合、授業内の教材を適宜調整する事で必要な前提知識を絞る事が重要となります。例えば、Java 言語によるアンドロイド アプリケーション開発の知識に不安がある場合、開発言語を Visual Basic 等に置換えたり、アンドロイドアプリケーションでは無く、クライアント・サーバ型の業務アプリケーション開発に置換るといった調整を行って下さい。また海外エンジニアの確保が困難な場合、日本人のみによる実施でも問題ありません。この場合、コミュニケーションを英語で実施するか、日本語で実施するかについても調整して下さい。

３．構成

３．１．カリキュラム全体の構成

カリキュラムの構成は、学習指導要領１．３カリキュラム全体の構成に記載されている通りです。学生が受講する順番としては、当カリキュラムの開発を実施する前に事前準備講座として、e-Learningによる当カリキュラムを受講するために必要な英語に関する知識とスクラムの基本を学習します。次に当カリキュラムの流れを説明会を通して通知し、更にスクラムシミュレーションを通して、スクラム開発手法の理解を深めます。

事前準備講座を終了した後に、開発環境の構築として、クライアント環境の構築を実施して、既に講師によって構築されているサーバ環境への接続とサーバ環境の説明を実施します。これらの作業が終了した後に、開発実習として３回のスプリントを実施します。

３．２．事前準備講座 e-Learning

e-Learning に関する授業展開の詳細については、e-Learning のシラバスを参照して下さい。また講師自身も事前準備として、当該 e-Learning を受講されることを推奨します。この e-Learning を受講することで、当システム開発時に必要な英語知識を効率良く習得することが出来ます。内容としては、スクラム開発で使用される専門用語の使い方や、e-mail 等を利用した開発を行う際に必要な英文の作り方、プレゼンテーションを英語で実施する場合のノウハウなどをスクラム開発手法に特化してまとめられています。またスクラム開発手法に関する基礎知識を日本語と英語で習得できるように工夫されています。

３．３．カリキュラム説明会及び、開発手法の理解、受講環境の構築と各種ツール操作の習得、ユーザー要件の説明、スクラム入門（第１週～第２週：全２週間）

当カリキュラムの目的と実施内容を学生に理解させることから始まります。教材は、“1.20_受講にあたって.pptx”を使用します。また、e-Learning の実施状況を確認し、未完了の学生はこの第１週～第２週までに完了するように指示します。

この週までの間に、スクラムシミュレーションを使用したスクラム実習を実施します。教材は、

“2.10_Scrum_Simulation1_Introduction_jpn.pptx”及び、“2.20_Scrum_Simulation2_Practice_jpn.pptx”を使用します。このスクラムシミュレーションを実施することにより、スクラム開発手法を実施する際のエッセンスを学習できます。スクラムシミュレーションの実施に不安がある場合は、有償のスクラム実習講座を予め受講される事をお薦めします。有償の実習講座として、ワイクル株式会社が実施している“LEGO(R)ではじめるスクラム入門（レゴスクラム)”を推奨します。

講師が行う事前作業としては、Web サーバの構築と Redmine サーバの構築があります。教材は、“4.30_テスト用サーバ環境構築手順書(WindowsOS 版).docx”及び、“4.40_Redmine-Backlogs サーバの構築手順書.docx”、“4.50_Redmine 環境設定(BackLog).pptx”を参照して下さい。

また Web サーバをインターネットに公開して利用する場合は、上記 Web サーバ構築手順を参考に CentOS など Linux OS による Web サーバ構築をお薦めします。

３．４．スプリント１の演習（第３週～第６週：全４週間）

このスプリント１は、実際の開発としての最初の４週間にあたり、学生がはじめて本格的なスクラム開発手法を実施する非常に大事な演習期間になります。教材は、“1.50_スプリント実習　課題説明.pptx”です。ほとんどの場合、プランニングしたバックログの全てを完了する事は出来ません。その為、スプリントの途中でバックログの優先順位上、必ず完了させるバックログをチーム内で話し合って、必ず完了できるようにチーム全体で協力し、遂行して下さい。もし、リスケジュールしたにも関わらず完了出来なかった場合は、スプリント１の期間を延ばして出来るだけ完了する様にして下さい。(本来のスクラム開発では期間を延ばすことはしませんが、授業としてプランニングの正確さと責任感を高めるためにこのような手段を講じて下さい。但し、スプリント２以降はスケジュールを延長する事は行いません。)

また、スプリントレビューとレトロスペクティブの実施結果を講師とともに、確認する時間を設けて、必要な情報が記述されているかと、書き方に問題が無いかを検証します。スプリント１の目的は、スクラム開発の手順と記載しなければならないドキュメントの記載方法と内容に問題が無い事の確認を第一の目的として実施します。

スプリント１は、日本人のみでの開発実施をお薦めし、海外エンジニアはスプリント２からの参加を推奨します。理由としては学生において、はじめてのスクラム開発手法での開発において手法もままならないまま、英語と海外エンジニア対応が加わると、負荷が高くなりすぎると予想される為です。

３．５．スプリント２の演習（第７週～第１０週：全４週間）

スプリント１で学んだスクラム開発手法の手順に従った開発を本格的に実施するフェーズとなります。特にディリースクラムで報告される問題点に注意して下さい。ここで述べられた問題点をそのまま放置せず、スクラムマスター（または講師）がディリースクラム終了後に関連するメンバーを集めて対応策を検討し、対応策を実施するようにして下さい。また対応策の実施結果をディリースクラムにて経過を必ず報告させて下さい。

またスプリントプランニングにおいて、スプリント１での開発実績を参考により精度の高いプランニングを行い、できるだけ予定通りに開発作業を完了出来る様に進めて下さい。

このスプリントから海外エンジニアの参加を推奨します。海外エンジニアについても、日本の学生

と同様な事前学習を受講していることを前提とします。第１回目の協働打合せの日時と方法(TV 会議やチャットなど)は、予め講師側で調整し学生に通知して下さい。２回目以降の打合せについては、チーム内で調整していくものとします。

このスプリントでは、アプリケーション開発に加えて海外エンジニアを含めたチーム内のコミュニケーションを円滑に行うことを重視し実施します。スプリント１と同様にディリースクラムで報告された問題点を放置しないで、チーム内で出来るだけ早く対策を実施できるように注力させて下さい。

またこのスプリントからは、プロダクトオーナーの役割がより重要になってきます。開発の進捗状況や発生している問題点などの情報を考慮して、柔軟にバックログの優先順位と開発対象機能の調整をチームとの打合せを通して決めて、開発を進めていくように指示して下さい。

３．６．スプリント３の演習（第１１週～第１４週：全４週間）

このスプリントが最終スプリントになります。しかし、スプリント１の延長などの理由から短縮したり、実施できない可能性も考慮して下さい。いずれの場合でもこのスプリント終了時に顧客に最終的に開発されたアプリケーションを受け渡せるように準備させて下さい。当スプリント終了時のレトロスペクティブにおいては、もし次のスプリントを実施する場合にどのような機能を開発対象とするかについてもチーム内で打合せた結果を記載させて下さい。

このスプリントで開発作業がクローズされるので、ここまで予定と実績の集計を行って下さい。具体的には、各スプリント毎にどのような機能実装(要件定義の要件をベースとします)を予定し、開発とレビュー、リリースの実績がどうなったかをまとめます。実績としては、機能-バックログ毎の作業時間数、レビュー結果、リリースの可否を主項目とし、補助項目として開発したソースの追加・修正ステップ数や、データベーステーブルの変更状況、テスト計画及び実施結果などをまとめて、報告資料を作成します。

３．７．成果発表及び、最終評価（第１５週）

実際の開発であれば、スプリント終了＝プロジェクトの終了であるが、このフェーズは、学習としてのまとめとして必要です。教材は、“1.60_成果発表　課題説明.pptx”を使用します。

基本的にこれまで行ってきた開発作業に関する成果をとりまとめて、成果発表のプレゼンテーションを実施します。最終プレゼンテーションは、可能であれば英語で実施する事を推奨します。この英語による資料作成及び、発表経験も学生の意欲向上と自信につながります。

また講師が、個々の学生が行ってきた作業に対する評価を実施し、評価結果を総評を加えて学生に通知する事を目的とします。

最後にチームとして、スクラム開発手法によるアプリケーション開発をスプリント毎に振り返り、より良く開発を実施する為に、どうすればもっと良くすることが出来たかを話し合って下さい。

４．おわりに

当カリキュラム及び教材は、スクラム開発手法と海外エンジニア及びチーム内のコミュニケーションをより良く行えるための方法を学習することに主眼を置いています。

その為、カリキュラムを実施する学習環境に応じてカスタマイズされる事をお薦めします。例えば、受講者が Java 言語学習を行う予定が無い場合や、アンドロイドアプリケーション開発を実施する予定が無い場合は、サンプルアプリケーションを VisualBasic 言語や C#言語によるクライアントサーバ型システムに変更したり、Web やインターネットに特化した学部・学科で実施する場合は、HTML5 による HTML コンテンツをサンプルとして使用する事で、スクラム開発手法を学習することも出来ます。

また、純粋にスクラム開発手法による開発を学習するための教材として、すべて日本語での開発を実施しても学習効果を得ることが出来ます。

更に、システム開発を学習しない学部・学科に対して、スクラム開発技法の学習教材として使用する場合は、当カリキュラムの事前学習教材である e-Learning とスクラムシミュレーション実習のみを使用する事で学習効果を期待できます。

これらのように、当カリキュラム及び各種教材それぞれの意図と利用目的を正確に理解することで、様々な学習者に合わせてカスタマイズして利用することが出来ると考えています。

以　上

平成２７年度　産学連携による実践型人材育成事業

富山県をモデルとした地方型グローバルＩＴエンジニアの

育成カリキュラムの構築推進プロジェクト

富山県をモデルとした地方型グローバルＩＴエンジニアの育成教材

発行 平成２８年 ２月

編者 学校法人 浦山学園富山県をモデルとした地方型グローバル
ＩＴエンジニアの育成カリキュラムの構築推進プロジェクト実施委員会

連絡先 〒939-0341 富山県射水市三ヶ５７６
電話 0766-55-1420　Fax 0766-55-0757