

みんなで Scrum!!!!

山科 和史

CONTENTS

1	はじめに.....	2
2	アジャイル開発手法とは？.....	2
3	Scrum.....	3
3.1	Scrum の定義.....	3
3.2	Scrum の理論.....	3
3.3	Scrum フレームワーク概要.....	4
3.4	スプリントの実践方法.....	6
3.4.1	スプリントプランニング.....	7
3.4.2	デイリースクラム.....	9
3.4.3	スプリントレビュー.....	11
3.4.4	スプリントレトロスペクティブ.....	12
3.4.1	バックログの項目について：粒度の決め方例.....	13
4	理解度チェック.....	15
5	さいごに.....	15
6	Reference.....	15

1

1 はじめに

こんにちは。LED-Camp 4 実行委員の山科和史（やましなかずし）です。本資料では LED-Camp4 のカリキュラムで行う予定の、「みんなで Scrum!!!!」における前提知識となる Scrum フレームワークについての基本的な知識をまとめております。ぜひご一読ください。なお、本資料は Scrum フレームワークを作り上げたことで有名なジェフ・サザーランドさんの **The Scrum Guide** [1] に準じた資料になっています。

2 アジャイル開発手法とは？

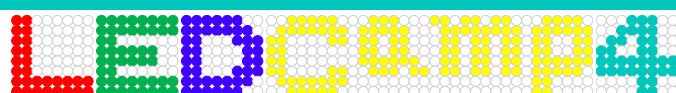
アジャイル開発では開発を行う対象（プロダクト）を複数の機能に切り分け、比較的短い単位期間において各機能を開発します。それゆえに常に**利用可能なプロダクト**を開発しつづけるという特徴があります。また、機能仕様の変更などに柔軟な対応が可能です。各単位期間の終了後は「ふりかえり」を行うことによって開発における方向性は正しいかどうか判断し、以降の開発において軌道修正することも可能です。アジャイル開発手法はソフトウェア開発における開発手法の一群を総じて呼称するものであり、一つの開発手法に限定するものではありません。したがって、本稿において紹介する Scrum もアジャイル開発手法の一種です。以下に示すアジャイルソフトウェア開発宣言は 2001 年、当時ソフトウェア開発において名声のあった 17 人（ケント・ベックら）がアジャイル開発についてそれぞれの意見を議論し、結果をまとめたものです。

アジャイルソフトウェア開発宣言 [3]

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

プロセスやツールよりも**個人と対話**を、
包括的なドキュメントよりも**動くソフトウェア**を、
契約交渉よりも**顧客との協調**を、
計画に従うことよりも**変化への対応**を、
価値とする。すなわち、左記のことがらに価値があることを
認めながらも、私たちは右記のことがらにより価値をおく

事前学習テキスト：みんなで Scrum!!!!



3 SCRUM

3.1 SCRUM の定義

Scrum は**変化の激しい問題に対応するためのフレームワークであり、可能な限り価値の高いプロダクト開発を行うためのものです**。特に Scrum は以下のような特徴を持ちます。

3

- 軽量
 - 覚えることは多くない
- 理解が容易
 - やってみるのは簡単
- 習得は困難
 - 説明・継続するのが難しい

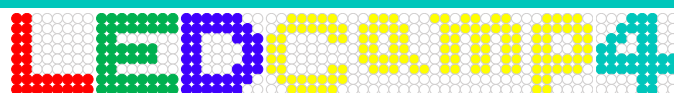
Scrum は本節以降で説明する、Scrum チームにおける役割、各イベント、作成物、ルールで構成されます。これらはそれぞれ目的があり、スクラムの利用には欠かせないものです。

3.2 SCRUM の理論

Scrum では経験主義的な工程（プロセス）制御を行います。**経験主義**とは、実際の経験や既知に基づく判断によって知識を獲得できるという考え方です。したがって、Scrum ではフィードバックの機会をより効果的にするために反復的にふりかえりをして、プロダクト開発におけるプロセス制御の最適化、リスクの管理を行います。特に、経験主義に基づくプロセス制御の実現は以下の要素によっておこないます。

- 透過性
 - 最終目標や進捗、解決すべき問題をチーム内の全員が共有し、見える化する
- 検査
 - Scrum 開発における作成物や進捗を検査し、目標への方向性を確認する
- 適応
 - 進捗の著しい遅れや、成果の不備が確認された場合、素早く調整・改善を図る

事前学習テキスト：みんなで Scrum!!!!



このような要素を反復的かつふりかえりながら実践することで常に動作し、利用可能なプロダクトを提供します。各要素が Scrum においてどのように実践されるかに関しては 3.3 節以降において説明します。

3.3 SCRUM フレームワーク概要

4

図 1 に Scrum フレームワークのモデルを示します。

Scrum フレームワークでは初めに顧客からの要求を解析し、**プロダクトバックログ**を作成します。プロダクトバックログは要求リストのようなもので、このリスト中にある項目がすべて達成されるとプロダクトの完成となります。各スプリントでは最初にプロダクトバックログの中において、各要求に対して適切な価値づけと現在のスプリントで何を行うかの選択を行います。

次に、選択した要求に対してスプリント内に要求の実現をするためにはどんなことをすべきかタスク切り分けを行います。この工程において発生したタスク群を**スプリントバックログ**と呼び、選択されたプロダクトログの項目を**完了**するために必要な作業リストになります。

スプリントバックログをもとに実際の開発をスタートし、実装を済ませ、**完成（チームメンバー内で共有している定義において）**したものは**プロダクトバックログアイテム**と呼びます。また、各スプリントにおいて完成してきたプロダクトバックログアイテムのまとまりを**インクリメント**と呼びます。インクリメントは常に利用可能であり、動作しなければなりません。さらに各スプリントの終わりには、今回のスプリントにおいて完成したプロダクトバックログアイテムを追加した**新しいインクリメント**が完成します。

スプリントの最後には**レビュー**を行います。これはインクリメントの検査をしたり、プロダクトバックログの再検討をしたりします。また、レビューを終了した後に**ふりかえり**を行い、今回のスプリントにおいてチームメンバー同士の関係性やプロセス制御の具合などの観点からスプリントをふりかえり次のスプリントへの改善計画を立てます。

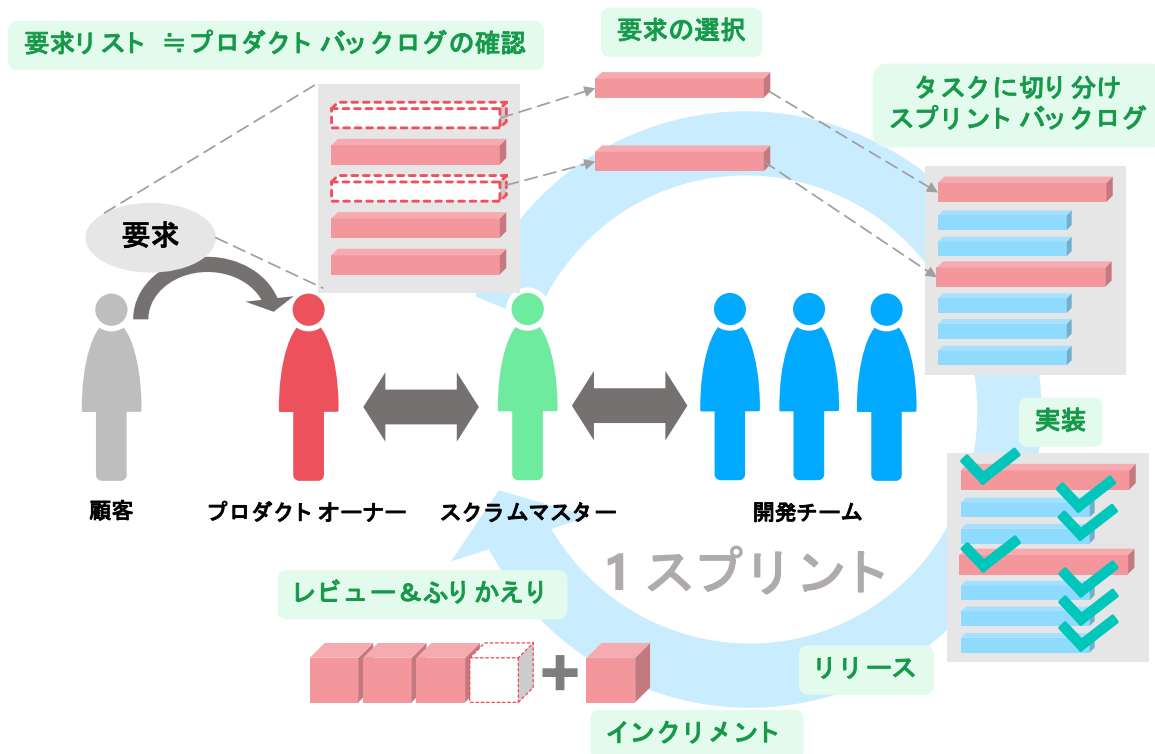


図 1 Scrum フレームワークのモデル

Scrum におけるプロジェクトは**スプリント**と呼ばれる比較的短く、決められた時間割を繰り返すことによって行います。また各役割を以下に示します。

プロダクトオーナー

開発するプロダクトの責任を持つ人。顧客からの要求を分析し、開発チームへ適切に伝える役割を持つ。特に、**プロダクトバックログ**の管理を行い、開発チームが行うべき作業の価値を最適化し、目的を達成できるようにする。なお、プロダクトバックログの管理自体は開発チームが行う場合もある。

スクラムマスター

Scrum についての理解と成立に責任を持つ人。チームに指示や命令をする役割ではない。すなわち、チーム（開発チーム、プロダクト）が適切に Scrum を成立させるために守るべきルールを提示し、支援する。また、チームが自主的に問題解決できるように促す。

開発チーム

実際に開発を行う人。常に、自己改善、チームの改善を念頭におく。開発チームでは各スプリントにおいて、「完成」したプロダクトインクリメントを作り上げる。

3.4 スプリントの実践方法

Scrum フレームワークにおいて、あらゆるイベントは**上限のある時間枠**を持ちこれを**タイムボックス**と呼びます。また、基本的に一度決定したタイムボックスは変更することはありません。スプリントは「完了」した動作するプロダクトインクリメントを作るための期間であり反復する単位です。Scrum におけるスプリントは通常、1 か月以内の範囲というタイムボックスで形成します（LED-Camp では数時間単位）。スプリントが長すぎると開発対象の定義が変更したり、リスク増大したりする可能性があるためです。スプリントにおいて定める目標を**スプリントゴール**と呼び、スプリント内ではこのゴールに悪影響を及ぼすような変更を加えてはいけません。またプロダクトオーナーのみがスプリントの中止権限を持ちます。

6

図 2 にスプリントの流れを示します。スプリントを構成するイベントは順に下記のものがあります。

- **スプリントプランニング**
 - スプリントの計画ミーティング
- **デイリースクラム**
 - 開発における進捗状況と次やることの確認
- **スプリントレビュー**
 - インクリメントの検査、プロダクトバックログの再確認
- **スプリントレトロスペクティブ**
 - 今回のスプリントのふりかえりと次の改善策を検討

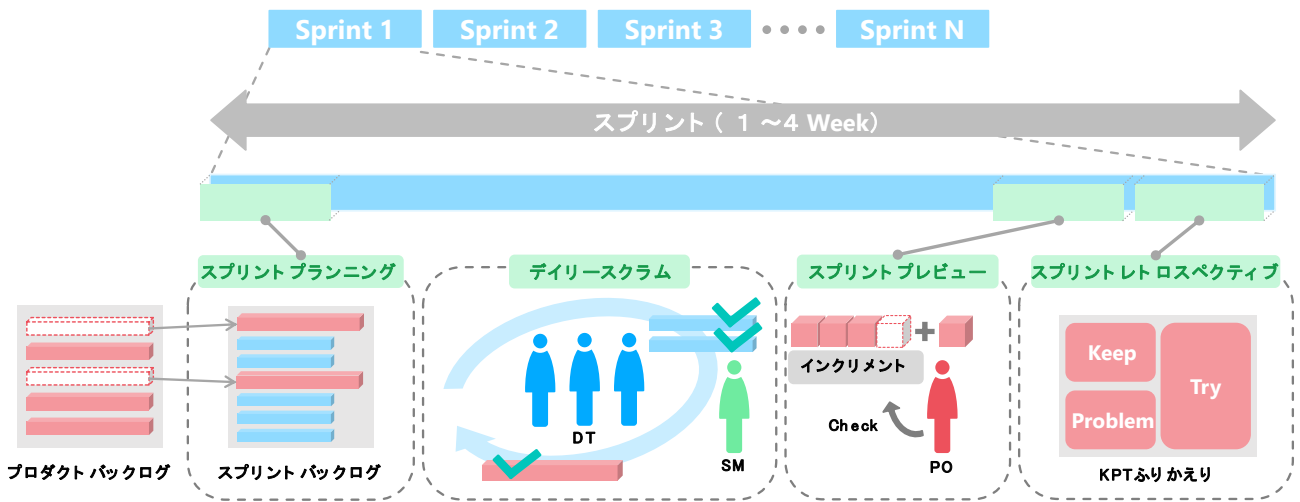


図 2 1スプリントの流れ

3.4.1 スプリントプランニング

スプリントプランニングとは各スプリントの始めに行い、現スプリントで遂行すべき作業を計画するミーティングのことです。スプリントプランニングでは大きく分けて2つのトピックについて話し合います。

Topic 1 : スプリント内に何を開発するか (プロダクトバックログから選択)



図 3 プロダクトバックログ

開発チームはプロダクトバックログからこのスプリントにおいて実装すべき機能を検討します。プロダクトオーナーはこれをもとにスプリントにおいて達成すべきゴール (スプリントゴール) を設定します。この際に重要になるのが図 3 に示すようなプロダ

クトバックログに示される要素たちに対する価値ポイントと相対見積りポイントです。相対見積りポイントは1スプリント内に開発が可能かどうかの、相対的な見積り値であり、価値ポイントはプロダクトにおいて、対象の機能はどの程度必要かを表す値です。プロダクトバックログでは各項目に対して2つのポイントが適切に設定されなければなりません。そこで適切な見積りポイントと価値ポイントを定めるための方法の一種として**プランニングポーカー**があります。

8

プランニングポーカー

- チームメンバが 1, 2, 3, 5, 8, 13 といったカードを持ちプロダクトバックログの項目やタスクに対してカードを出す
 - 数字がそろった場合：適用
 - 数字が合わない場合：Max と Min の人が話し、再度全員がカードを出す
 - ◇ Max を出した人：難易度・リスクを知っている
 - ◇ Min を出した人：スマートなやり方を知っている
 - さらに合わない場合はあらかじめ決めたルールに基づき決定（例：Max 値を採用）

Topic 2 : 対象をどのように開発するか (スプリントバックログ : タスク切り分け)

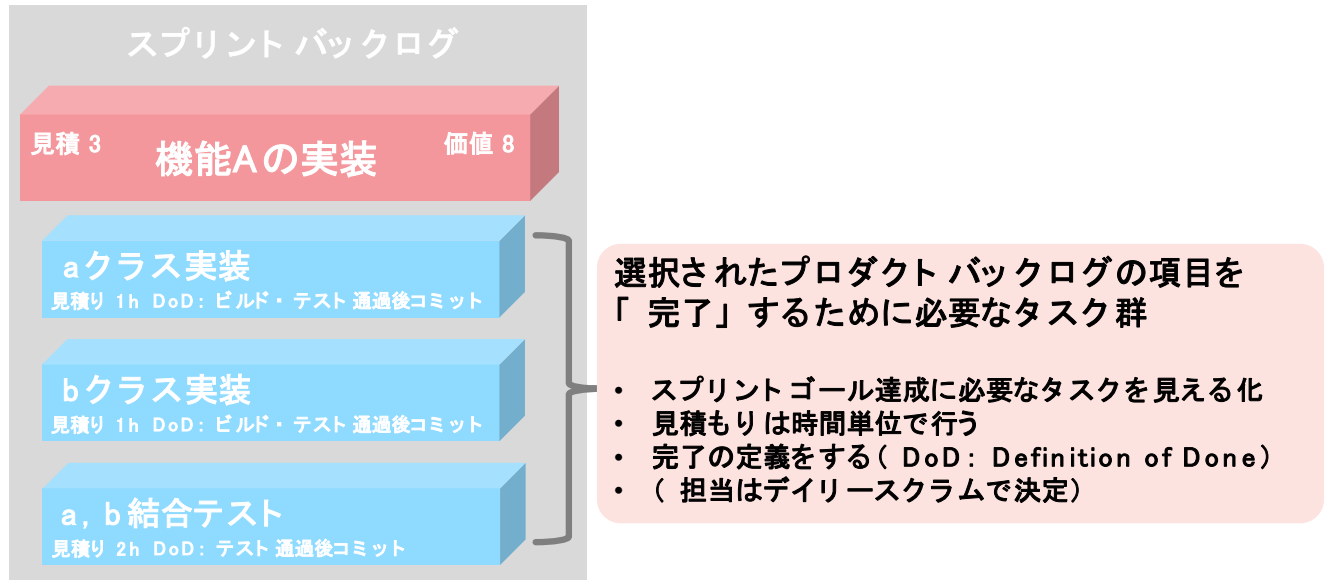


図 4 スプリントバックログ

プロダクトバックログから機能 (プロダクトバックログアイテム) を選択し、**スプリントゴール**が決まると次は、スプリント内でどのように機能を開発するかを決めます。この際、機能を実現するためにどんなタスクが必要かを抽出し、**スプリントバックログ**にリストアップします。この際、タスクがどの程度の時間で遂行できるかの見積もりと、タスクの完了定義を行います。特に、**完了の定義 (Definition of Done)** は**チームメンバ全員が共有しなければなりません**。

3.4.2 デイリースクラム

デイリースクラムは開発チームの状況確認後、つぎの 24 時間の計画を作るためのイベントです。きわめて短い時間において話し合いを終えることがルールであり、以下の 3 つの質問に答えます。

- 前回のデイリースクラムから行ったこと
- 次回のデイリースクラムまでに行うこと
- 問題点
 - デイリースクラムにおいては問題抽出のみ行い、対策は別途行う

開発における進捗管理の方法とツール

開発におけるチーム全体の進捗を管理、見える化するためのツールを紹介します。また各ツールはどんなものか図 5~7 に例を示しています。

タスクボードとチケットシステム

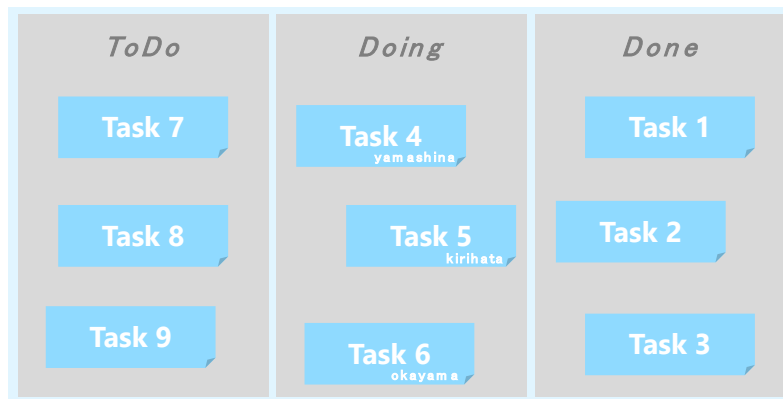


図 5 タスクボードとチケットシステムの例

- 現在、残っているやるべきこと、完了したことを確認できる
- 今誰が、何に取り掛かっているかがわかる
- タスクとして貼り付けるのはスプリントバックログにおけるタスク

リリースバーンダウンチャート：プロジェクト全体の進捗の俯瞰

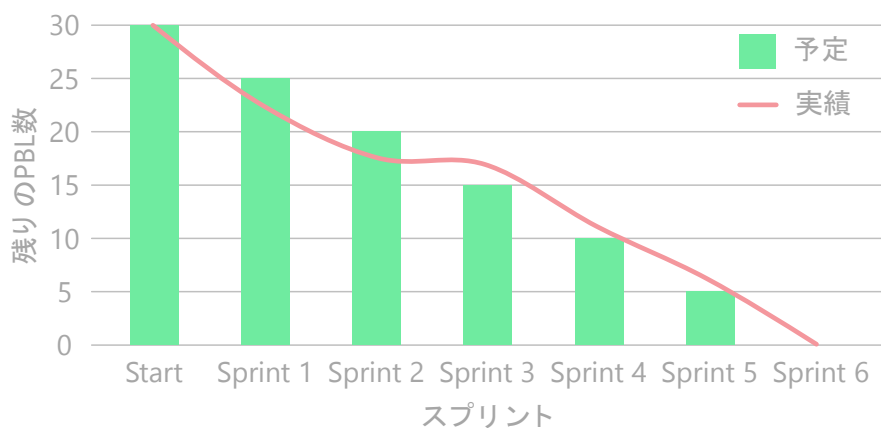


図 6 リリースバーンダウンチャートの例

- Sprint 毎のプロジェクトバックログの難度の合計の見積りと実績をプロットしたもの
- 傾きの平均値を Velocity と呼び、チーム開発力の指標となる
- Velocity が見える化できると、以降の開発の見積りが、より正確になる

スプリントバーンダウンチャート：スプリントの進捗の俯瞰

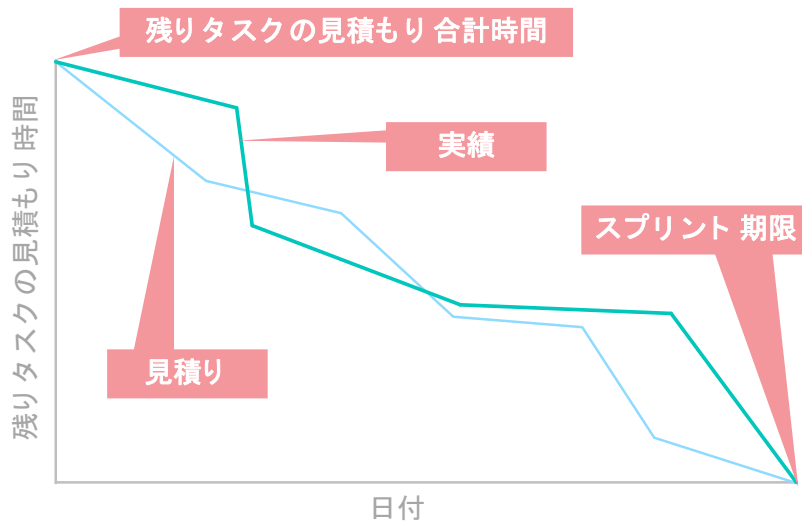


図 7 スプリントバーンダウンチャートの例

- Sprint 毎に残りタスクの見積もり時間において作業見積りと実績をプロットしたもの

3.4.3 スプリントレビュー

スプリントレビューとは、スプリントの終わりにインクリメントの検査と、必要であればプロダクトバックログの適応を行います。具体的には、プロダクトオーナーは何が「完了」し、何が完了していないかを特定します。また、開発チームはインクリメントのデモを行い、動作し、利用できる状態であることを示します。インクリメントのデモを通じて、フィードバックを引き出し、必要に応じてプロダクトバックログの修正を行います。

3.4.4 スプリントレトロスペクティブ

Keep

効果があったため、
引き続き行うこと

Try

次回から挑戦すること

Problem

問題があったため、
改善すべきこと

図 8 KPT のイメージ

スプリントレトロスペクティブでは Scrum チームの検査と次のスプリントの改善計画を立てるために行うふりかえりのイベントです。特に下記のような目的があります。

- 人・関係・プロセス・ツールの観点から今回のスプリントを検査する
- うまくいった項目や今後改善が必要な項目を特定、整理する
- Scrum チームの作業の改善実施計画を作成する

目的を達成するために、多く用いられるふりかえり方法に KPT（上図 8）があります。KPT は各メンバーが Keep、Problem、Try の三つの軸に沿ってふりかえりを行います。イベントにおいて Problem（タスクの遅延など）が出た場合、個人差を考慮し、チームとしてのサポート体制を改めて再検討し、改善案を出します。

KPT の重要性に関しては事前学習テキスト、

**「チームビルディングの理論と実践」の 4 章 “「ふり返し」と「分かち合い」”
をご参照ください。**

3.4.1 バックログの項目について：粒度の決め方例

Scrum フレームワークを行うにあたり、前項で説明したスプリントバックログとプロダクトバックログの項目の検討は、特に重要な作業になります。したがって、プロダクトバックログとスプリントバックログの項目を決める例を示したいと思います。

例えば飲み会の幹事になったとしましょう！

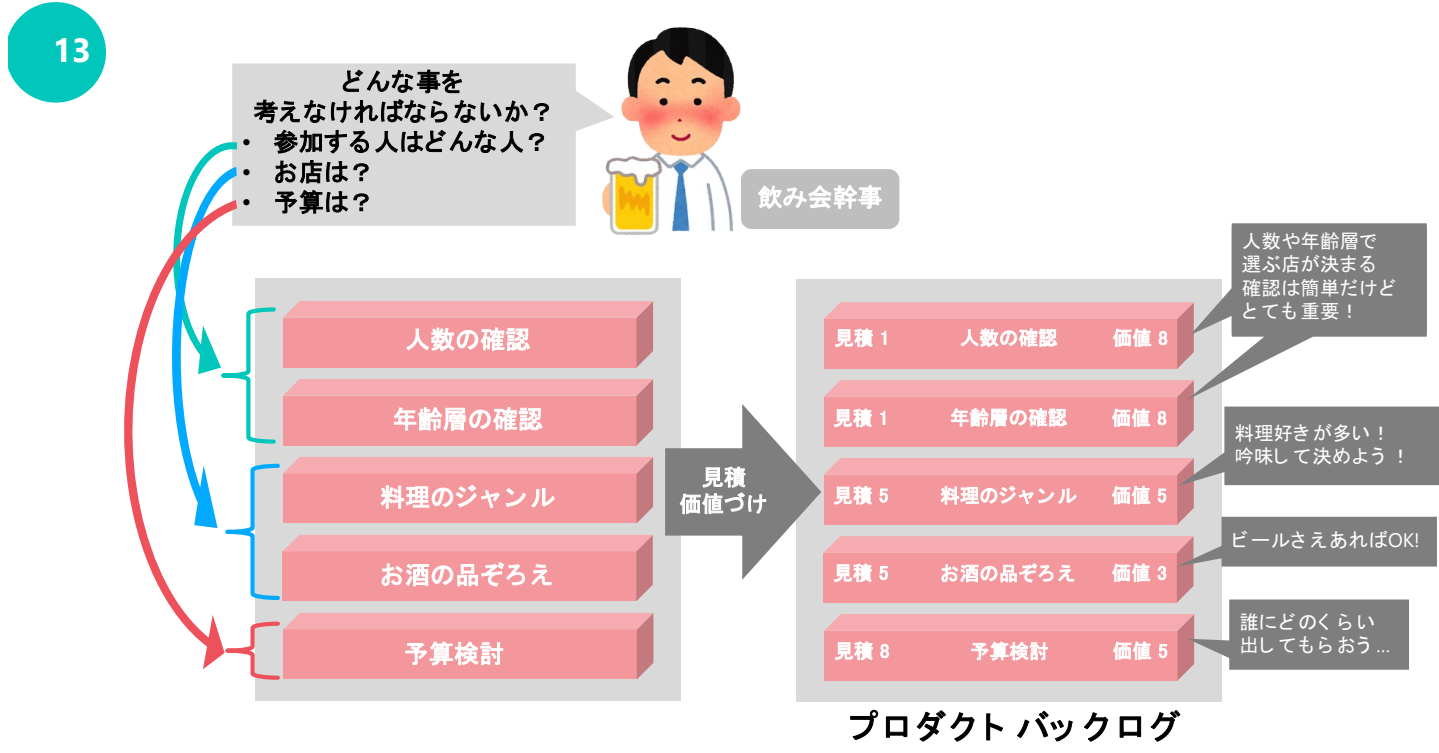
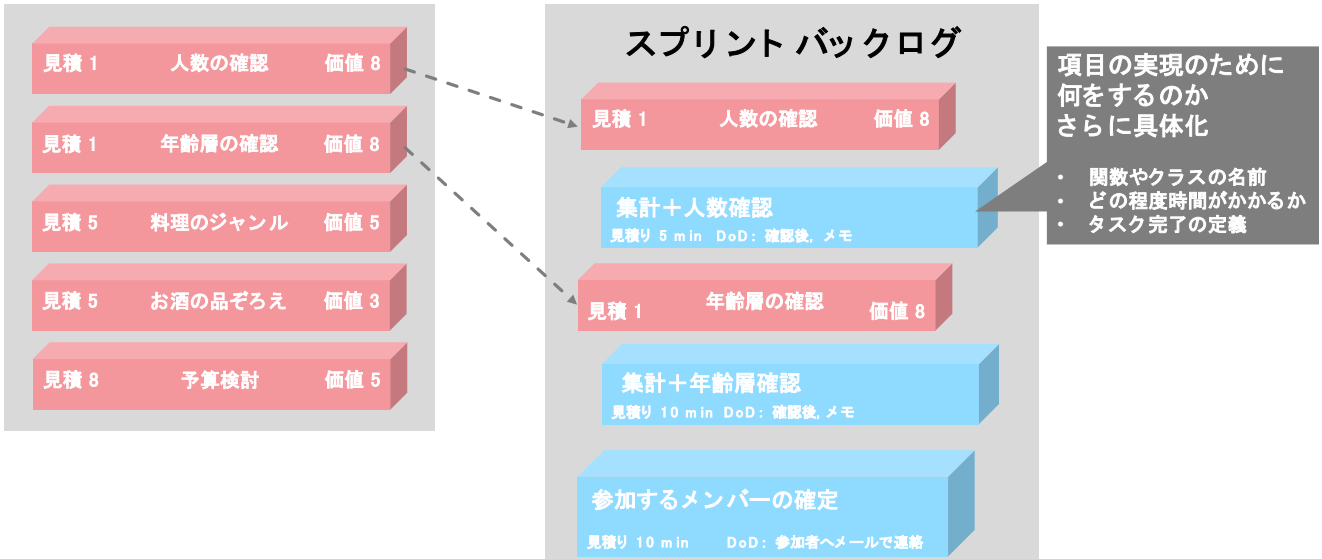


図 9 プロダクトバックログに検討すべき点を起こす

図 9 では、飲み会を開くにはどんなことを配慮すべきかを、分解して検討しています。その次に検討点を具体化し、価値付けを行うことで**プロダクトバックログの作成**を行っています。このようにプロダクトバックログの作成は要求や要件の分析・具体化をして、項目（プロダクトバックログアイテム）に起こす作業です。また、見積と価値づけについては、図中にあるように項目の実現の難易度と、飲み会を開くために必要な項目か否かという風に切り分けて考えます。例えば、抜きんでてお酒の品揃えのいいお店を探すのはなかなか大変ですが、「ビールさえ飲めればいい!」というならば、価値は低くなりますね。

また、プロダクトバックログで決めた各項目の優先度に応じて、各スプリントでスプリントバックログに項目のタスク切り分けをします。



スプリントバックログではスプリントで取り掛かる項目を、どのように実現するかを決め、リストアップします。特に、プロダクトバックログからさらに具体化し、何をしたらどのような成果が上がるかを見える化します。またタスクの切り分けの際はどの程度の時間がかかるかを見積もります。図 10 では分単位で書いてありますが、LED-Camp4 において、見積もり時間は基本的に時間 (hour) 単位で区切りましょう。

- このように最初は抽象度の高いところから徐々に具体化していき、最後は作業に結び付けていきます。
- プロダクトバックログとスプリントバックログに正解はありません。
- 各チームによって重要な事柄、タスクの切り分け方はそれぞれです。

4 理解度チェック

最後に理解度のチェックとして各項目の質問に答えてみましょう。

- アジャイル開発はどんなプロダクトを開発し続けますか？
- Scrum の定義はなんでしたか？
- Scrum フレームワークには大きく 3 つの役割がありましたがなんでしたか？
- Scrum フレームワークにおいて重要なバックログが 2 つありましたがそれはなんでしたか？
- ふりかえりには KPT を用います。それぞれの頭文字の意味はなんでしたか？

15

5 さいごに

ここまで読んでいただきまして、ありがとうございました。ご不明点、ご意見がございました場合はお手数ですが、LED-Camp のメーリングリストにご連絡ください。

led-camp4-sanka@swest.toppers.jp

6 REFERENCE

- [1] Jeff, Sutherland and Ken, Schwaber, ``The Scrum Guide``, <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-JA.pdf> - zoom=100, 2013
- [2] 細合晋太郎, ``みんなで Scrum!!!``, <http://swest.toppers.jp/LED-Report/Camp3/flashair/proceedings.html>, 2015
- [3] Kent, Beck and et all, ``Manifesto for Agile Software Development``, <http://www.agilemanifesto.org/>

事前学習テキスト：みんなで Scrum!!!

