

ソフトウェア工学

第14回

土田 隼之

週	授業内容・方法	週ごとの到達目標
1週	ソフトウェアの性質と開発	ソフトウェア開発の特徴および課題について少なくとも一つ上げられ、その理由を言える。
2週	ソフトウェア開発プロセス	複数の開発プロセスモデルを挙げ、それぞれの特徴を言える。
3週	要求分析	要求分析とプロトタイピングの関係性や有用性について言える。
4週	ソフトウェア設計	モジュールの結合度の低い場合と高い場合のモジュール間の依存性について述べ、モジュール結合度の低い具体例を言える。
5週	プログラミングとテスト	誤り混入をさせないためのプログラミング手法およびテスト効率を向上させる技法について言える。
6週	テストと保守	保守容易性を確保するための方策について、考察し、述べることができる。
7週	グループワーク	前半6週に関する課題を、グループワークで取り組む。
8週	中間試験	前半に習得した項目について確認する。
9週	オブジェクト指向 1	身の回りのモノに関して、クラスとインスタンスという言葉を用いて説明できる。
10週	オブジェクト指向 2	オブジェクト指向プログラミングの特徴について言える。
11週	ソフトウェア再利用	ソフトウェア再利用の重要性とその困難さについて言える。
12週	プロジェクト管理	プロジェクト管理の重要性を述べることができる。
13週	品質管理	品質管理手法について言える。
14週	ソフトウェア開発規模と見積もり	ソフトウェア開発規模の見積もり手法について言える。
15週	グループワーク	後半6週に関する課題を、グループワークで取り組む。
16週	期末試験	後半に習得した項目について確認する。

アジャイル開発と見積もり

模擬試験を予定してます
#中間前の模擬試験同様に
スライド見ながら解いて良い

今日の内容

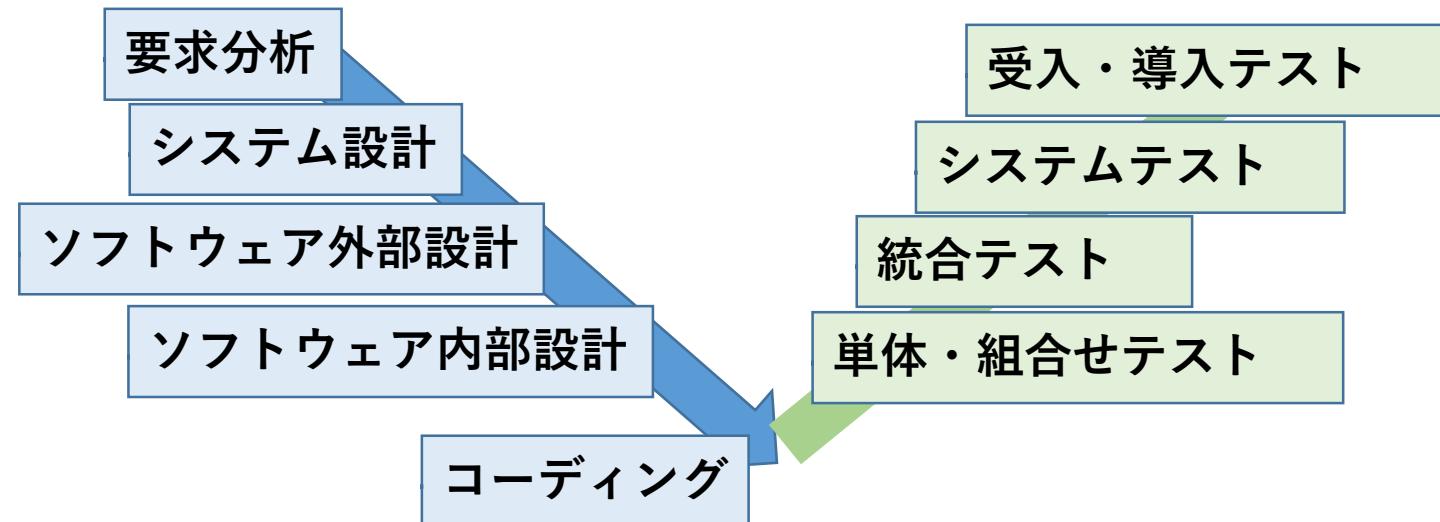
1) 「ソフト開発の難しさ」とアジャイル

2) アジャイル開発とは

プロセス定義

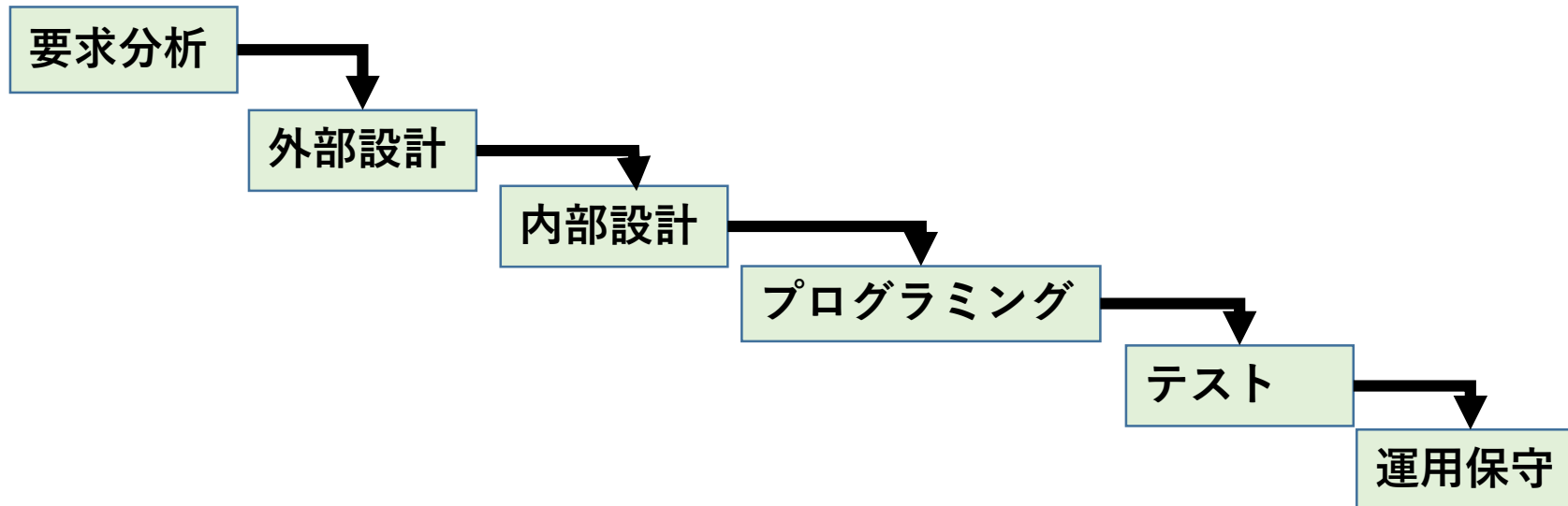
開発作業を効率的に遂行し、管理するための作業の骨組みを定めることをプロセス定義という。プロセスは、大まかな開発段階(フェーズ)分けと各段階での成果物が設定されている。

通常、各企業は組織内にいくつか標準プロセスを持っており、開発するソフトウェアの性格に応じて仕立てて使用することが多い。例えば、ウォーターフォールモデルはよく使われるモデルの一つである。管理プロセス例としては、設計と試験を対応付けたV字型モデルやスパイラルモデルがある。



2) ウォータフォールモデル

- 開発の各段階を逐次的な手順で進める方式
- 要求分析を明確にしてから外部設計、内部設計など順番に行い、次に運用・保守にうつっていく
- 実際の開発で伝統的に行われているモデル



2) アジャイルモデル(スクラムモデル)

• 構成要素

1) プロダクトバックログ: 実現したいことをリストにして並び変える

#定期的にメンテナンス。実現したい順に並び変える

2) スプリント: 1か月までの同じ期間に区切って繰り返す。

例: 2週間、2週間など。一つの区切りをスプリントと呼ぶ

3) スプリントプランニング: スプリントで何を達成するかを決める

#バックログのどれを、次のスプリントまでにどこまでするか
バックログをどうやって実現するかタスクに分割する

→タスクは、一日で終わりそうなサイズにする

バックログ項目を特定の担当者に紐づけることはしない

4) デイリースクラム: ゴールの達成に向けて進んでいるか毎日点検する(1日/15分とか)

• 構成人員

プロダクトオーナー(1名): プロダクトの価値を最大化

プロダクトバックログの管理者

開発チームに相談できるが干渉できない

開発チーム(3~9人): ソフト開発を行う

肩書やサブチームは無し

「ソフト開発の難しさ」とアジャイル

ソフトウェア開発での主な難しさに以下があるが、アジャイルである程度対応可能

1) ユーザの要望を把握するのが難しい

発注元の担当者にヒアリングを行い、「ソフトウェアへの要望」を聞き出す必要がある。

→開発要望は、発注元の担当者の頭の中にあるが、引き出すのは容易ではない。「**発注者の回答**」と「**実際の要望**」に差があるケースが多い。

アジャイルだと早めに現物(動くソフト)が見せられる

2) 詳細な仕様を決めるのが難しい

要望例:二つの数値を入力して、「計算」ボタンを押下した後に、二つの数値の積を表示

→**単純な要望に見えるが、ある程度開発が進まないと詳細は見えにくい。**

例1)どちらかに数値を入れず「計算」ボタンを押した場合？

例2)入力する数値の範囲は？

例3)計算後に入力した数値は消すのか？消さないのか？

アジャイルだと、適切な開発時期に仕様詳細を決められる

「ソフト開発の難しさ」とアジャイル2

ソフトウェア開発での主な難しさに以下があるが、アジャイルである程度対応可能

3) 正確な要求を相手に伝えることが難しい

ウォーターフォールでは、要求定義から完成品にたどり着くまで複数の工程を実施する必要があり、受入テストで発覚した仕様漏れは開発に大きな影響を与える。

→ **要求通りに正しく設計・動作できているかの確認が最後の工程**まで判断できない。正しく要求を伝えるためには、**実現方法のみを伝えるだけではなく、その背景や課題、目的、方針などを伝えることも重要**となる。

アジャイルだと開発方針や目的をチーム内で共有しやすい

4) 品質の確保が難しい

「2入力の積」のような単純な要求でも、考えるべき条件は多々存在する。どこまで考えるのかによって、設計・製造での仕様漏れの発生が増加や、テスト項目の増大が懸念される。

→ **品質を確保するためには、「テストの観点」を意識した設計および製造が必要となる。**

アジャイルだと「テストの観点」も仕様作成時に考慮される

アジャイルとウォーターフォール

「ソフトウェア開発の難しさ」の原因の一つは、「ソフトウェアが完成するはるか前からソフトウェアの完全な挙動を考える必要がある」というウォーターフォールの性質であると考えられる。

→「一通りの動作が行える小規模なソフトウェアを短いサイクルで開発・提供」するアジャイルでの開発では、ソフトウェア開発の難しさを緩和できる可能性がある。

ウォーターフォール開発と比較したアジャイル開発モデルの違いとして、以下がある。

1)短い期間の開発を繰り返す:アジャイルでは、固定された短い期間(例:数週間)で開発・確認ができるものを顧客にとって価値の高いものから順に製造・リリースおよび確認を反復する。この反復期間をスプリント(実装の期間)と呼ぶ。

2)優先度の高いものから作る:アジャイルでは、優先度の高い必須機能から作り、次にニーズの高い機能を追加開発する。必要最小限の機能実現を目指すため、当初予定した機能について作らないことも。

3)実現範囲を定めずに調整するスタイル:開発では、「実現範囲」と「QCD(品質、費用、納期)」をうまく調整する必要がある。実装すべきものを確認しながら開発するため、開発中に仕様変更など生じた場合は、それを受け入れて開発目標を柔軟に変更させるため、「実現範囲」を固定しない。

今日の内容

1) 「ソフト開発の難しさ」とアジャイル

2) アジャイル開発とは

アジャイルの概要

主要なアジャイル型開発手法の一つである「スクラム」での流れを扱う。

- 1) プロジェクトビジョンの共有(インセプションデッキの作成)
- 2) 業務全体像の理解(ユーザストーリーマッピング)
- 3) スプリント計画
- 4) スプリント
- 5) デイリースクラム
- 6) スプリントレビュー
- 7) ふりかえり

万能ではない。会議で意思疎通できないような規模の開発をやるのは適さない気もする。

1)プロジェクトビジョンの共有

アジャイル開発では、チーム全体が主体的に開発に取り組む必要があり、各メンバがプロジェクトビジョン(プロジェクトの目的や方針、背景)を共有しておくのが望ましい。プロジェクトビジョンの共有のため「インセプションデッキ」を作成する。

インセプション:プロジェクト全体の背景や目的、方向性、価値観についての認識合わせ

インセプションデッキ:認識合わせを端的なドキュメントで相手に伝える

プロジェクト開始時にしておくとい「いくつかの質問とその回答」のこと。

名称	説明
①我々はなぜここにいるのか？	プロジェクトの最終目標を決める
②エレベーターピッチ	相手を説得する文句を作る
③パッケージデザイン	製品のキャッチコピーを作る
④やらないことリスト	やること・やらないことを決める

．．．．．

1) インセプションデッキ①

①我々はなぜここにいるのか？→プロジェクトの最終目標を決める
開発プロジェクトの発足の経緯や目的を記述する。

なぜあなたはここにいるのか？

我々はなぜここにいるのか？

- ・大事な理由 その1
- ・大事な理由 その2
- ・大事な理由 その3

このプロジェクトを
行う最大の理由

- ・ 安価にPOSを導入したい
- ・ 注文処理の効率化
- ・ インタラクティブなメニューの提供(おすすめ)

< インタラクティブなメニューの提供 >

- ・ 「大事な理由 その1～3」

→企業の視点で、なぜこのプロジェクトにお金を使用するのかの理由

- ・ 「最大の理由」

その1～その3の中で、最も大事と思うものをピックアップして、具体的な理由を記載する。

1) インセプションデッキ②

②エレベーターピッチ→相手を説得する文句を決める

短い時間(15秒～30秒)の説明で相手の興味や関心を得るための売り文句を作る。

→エレベーター内でたまたま企業の経営者に会った時に相手を説得するための説明に相当

エレベーターピッチ

- ・[必要性や抱えている課題を解決]したい。
- ・[対象顧客]向けの、[プロジェクト名]は、[製品の 카테고리]です。
- ・これは、[重要な利点、対価に見合う説得力のある理由]があり、[他の主要な競合製品]と違って、[差異化の決定的な特徴]が備わっています。

オンライン英会話「レアジョブ」のエレベーターピッチ

- ・「**忙しくて英会話学習の時間を確保できない悩み**」を解決したい
- ・「**英語学習者**」向けの
- ・「**レアジョブ**」というプロダクトは、
- ・「**オンライン英会話**」です。
- ・このプロダクトは「**オンラインで本格的な英会話レッスンの受講**」ができ、
- ・「**対面式の英会話教室**」と違って、
- ・「**いつでも・どこでも、思い立ったときに利用できる仕組み**」を備えている。

1) インセプションデッキの作成

インセプションデッキは、それをベースに議論を行いお互いの理解を深めるために利用する。 特定の人を作って、他の人に共有するというやり方ではなく、**チームで集まって最初から一緒に作る**、あるいは一部のたたき台は準備しておき、チームで議論しながら作成する。

インセプションデッキは、作って終わりではなく、メンバ全員が見やすい位置に張り出す。そして、**プロジェクトは何を目的としているのか、そのためには何をやることにし、何をやらないと判断したのか、自分たちのプロジェクトのビジョンに対して、自分たちの活動がぶれていないかについて確認**し続ける。

課題14-1:締切8/1

就職先企業で新規製品の提案をする立場になったとして、以下考えてください。

#自分が好きな製品やサービスの下記インセプションデッキを作成するのも良いです。

#例:ポケモン(色は自由)のインセプションデッキは？

プロダクトオーナーの立場として、

「我々はなぜここにいるのか」「エレベーターピッチ」を作成して、隣の学生と共有して、感想を共有してください。

提出:作成頂いた資料(例パワポorPDF)、共有した感想

2)業務全体像の理解(要件の抽出)

アジャイル開発では、要件の抽出を「プロダクトバックログの作成」と呼ぶ。プロダクトバックログは開発初期から完璧な要件の一覧を作るものではなく、「スプリント」の作業を進めながら徐々に要求を明確な要件として確定していく。

バックログ:残務、積み残し、未処理分

プロダクトバックログ:製品(またはサービス)への要求の一覧”

実際に開発するタスク(プロダクトバックログ)を抽出するために、まずは「ユーザーストーリーマッピング」で業務の全体像を表現する。

ユーザーストーリーマッピング:システムの全体像を整理し、実現したい項目に優先順位をつけたプロダクトバックログを作成するための方法。

2) ユーザストーリーマッピングの作成

ユーザストーリーマッピングでは、**ユーザが取りうる行動を洗い出し、その行動を時系列に並べて、行動に基づいた「あるべき機能」を導き出す。**

そのため、現状の業務(As-Is)プロセスをあるべき業務(To-Be)プロセスとして業務内容を整理する手法に役立てられ、ユーザが最も大切にしている価値を重視し、システムの振る舞いがどのようになるかを表すことができるようになる。

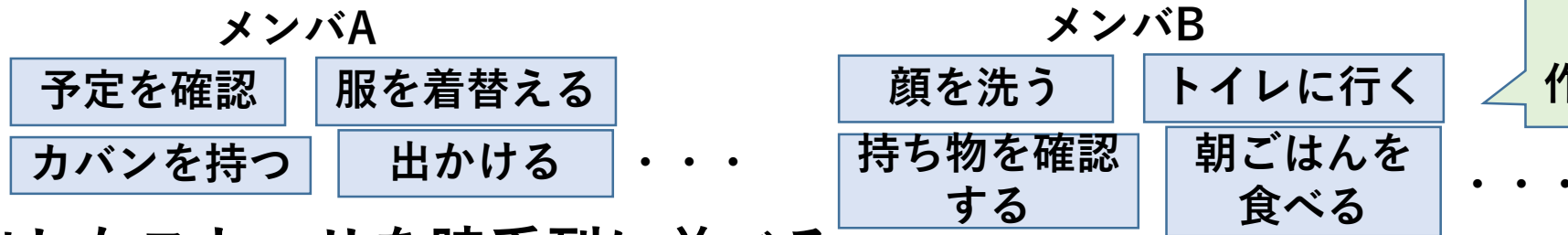
以下のような手順で作成される。

- a) 典型的な作業について、1項目につき付箋紙1枚に書き出す
- b) 書き出したストーリーを時系列に並べる
- c) ストーリーのタイトル(バックボーン)を付ける
- d) 例外的な事象とその扱いを考え、時系列に入れる
- e) ストーリーを優先度ごとに並べ替え、優先順位を明確にする

2) ユーザストーリーマッピングの作成

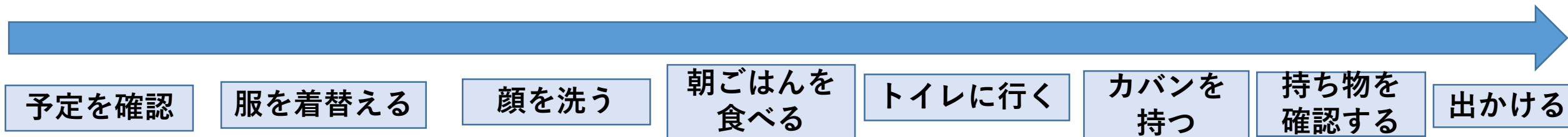
a) 典型的な作業について、1項目につき付箋紙1枚に書き出す

開発メンバ全員で、典型的な作業の流れを付箋紙に書き出していく。これを「ユーザストーリー」や「ストーリー」と呼ぶ。付箋紙には、作業の1項目につき、1枚の付箋紙を使う。この作業は個人作業として行い、他のメンバと相談して項目の洗い出しを行ってはいならない。「誰は(Who)」「何をする(What)」「理由(Why)」がわかるように書くのが望ましい。



b) 書き出したストーリーを時系列に並べる

各メンバで書き出したストーリーを出し合い、時系列の順に並べる。ストーリーの中で同じ時間内に起こるものがある場合は、上下にならべて配置する。ただ、作業上優先度の高いものは上側に配置する。

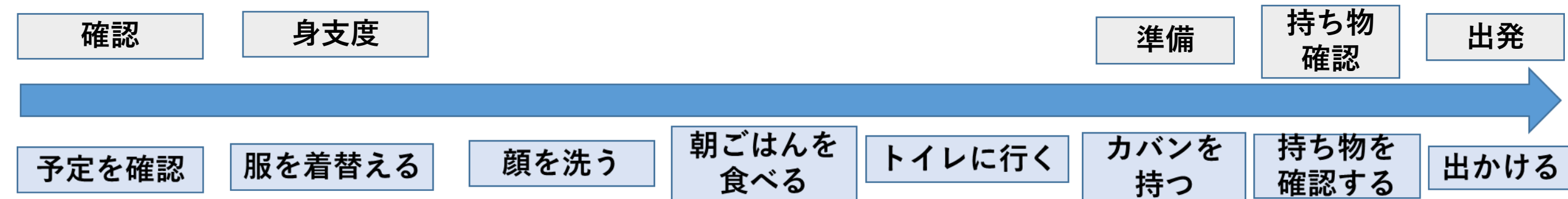


2) ユーザストーリーマッピングの作成

c) ストーリのタイトル(バックボーン)を付ける

同じ列または連続するストーリーについて、そのストーリーに共通したタイトルを付加する。
このタイトルは「バックボーン」と呼ばれる。

- ・ 予定を確認 → 「確認」
- ・ 服を着替える、顔を洗う、朝ご飯を食べる、トイレに行く → 「身支度」
- ・ バッグを持つ → 「準備」
- ・ 持ち物を確認する → 「持ち物確認」
- ・ 出かける → 「出発」

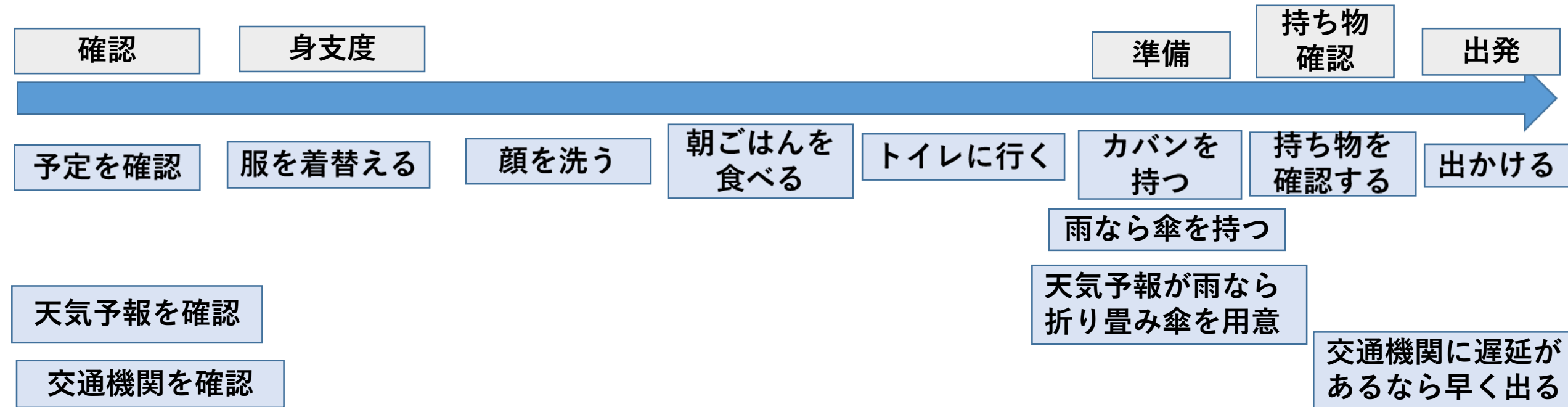


2) ユーザストーリーマッピングの作成

d) 例外的な事象とその扱いを考え、時系列に入れる

これまでに導き出したストーリーの流れの中で、「もし、～が起きたら」という例外的な事象の発生とその扱いを考える。

例外的な事象の発生	扱い
外出中に雨が降った	確認: 「天気予報を確認する」 準備: 「天気予報が雨なら折り畳み傘を用意」
交通機関の支障でセール開始時刻に遅れる	確認: 「交通手段を確認する」 出発: 「交通機関に遅延があるなら早く出る」



2) ユーザストーリーマッピングの作成

e) ストーリーを優先度ごとに並べ替え、優先順位を明確にする

最後に、ストーリーを優先度ごとに並べかえ優先順位を明確にしていく。優先順位が同じストーリーの集まりを作り、優先順位の高いものから順に上から下へ並べかえていく。**並べ替えを行ったときに、優先順位の境目がわかるように線で区切ってわかりやすくする。これを「スライスを切る」という。**スライスを切ったことによる優先順位の同じストーリーの集まりを「スライス」と呼び、上から順にスライス1、スライス2、、と呼ぶ。

確認

身支度

準備

持ち物確認

出発

予定を確認

服を着替える

カバンを
持つ

出かける

[スライス1:絶対に行う事]

雨なら傘を持つ

[スライス2:時間に余裕があったら行う事]

トイレに行く

持ち物を
確認する

天気予報を確認

[スライス3:もしものときのために行う事①]

天気予報が雨なら
折り畳み傘を用意

交通機関を確認

[スライス4:もしものときのために行う事②]

交通機関に遅延が
あるなら早く出る

課題14-2:締切8/1

進路決定(進学就職を決める、就職対策、編入対策)のストーリーマッピングを作成ください

手順1:まずは各自で作成ください

手順2:となりの学生と共有して、時系列に並べてみて各スライスを作ってみてください

提出:作成したスライスを提出ください

3) プロダクトバックログの作成

ユーザストーリーマッピングにより、優先して実装すべき作業項目が求められる。これをプロダクトバックログとして作成する。**プロダクトバックログは、優先順位の高いスライスから順に転記していく。**

「買い物に行く」という
作業なら、スライス1から順に転記

プロダクトバックログは、優先して確認・開発すべき要件を順に並べたリストである。プロダクトバックログは開発初期から完璧な要件の一覧をつくるのではなく、実際の開発工程である「スプリント」の作業を進めながら徐々に要求を明確な要件として確定していく。

開発初期の段階では、優先度の高いものから要件を明確にしていく。優先度の低いものは「その時点では要求・要件にあいまい性があるが実装に必要なもの」と考えている内容となる。優先度が低いものは、実際の開発工程である「スプリント」を進めていく中で、実装に必要であるのか、不必要であるのかを決定していく。最終的にすべての要件が導き出されることになる。

3) プロダクトバックログの作成

a) 優先順位の考え方

- ・ ユーザにとって最も価値が高いもので、プロダクトオーナーにより実現したい価値基準によって順位付けする
- ・ 早めに動作を確認しておかなければならないものには順位付けする

b) 1つの付箋紙に優先順位の異なる作業があった場合

- ・ この場合は、複数の作業に分割して再配置をするのが望ましい。

c) 適切なユーザストーリーマッピングの基本原則(INVEST)

- ・ 各ストーリーが以下を満たしているかを再確認する。

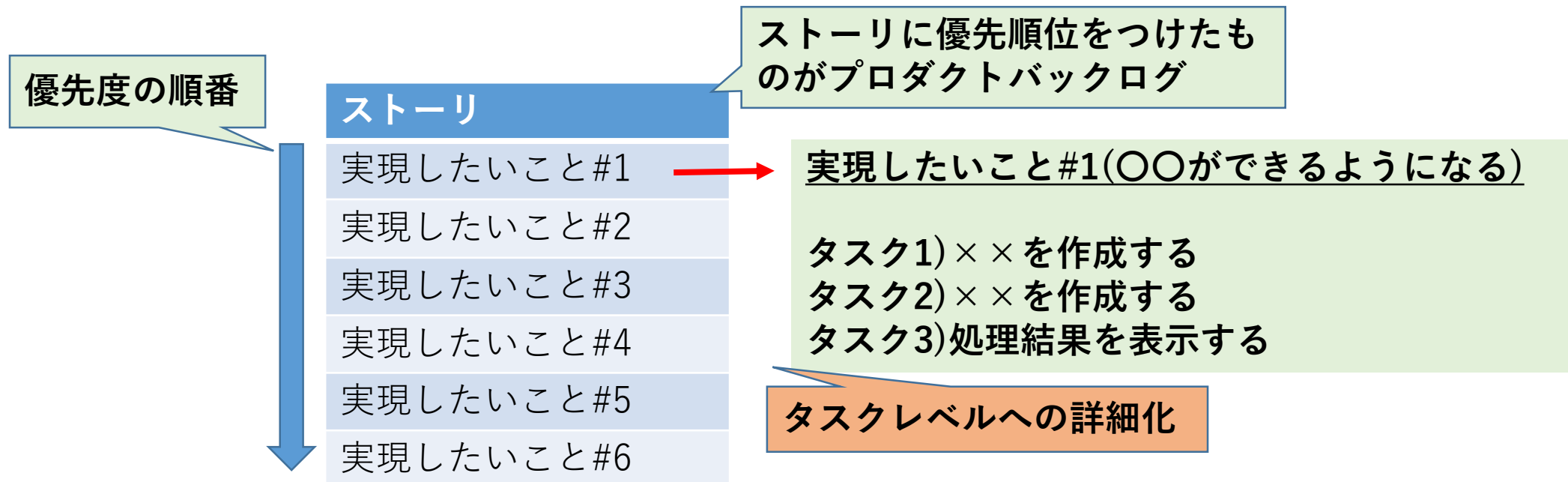
d) 非機能要求の明確化
→ 考慮すべき非機能要件も、付箋紙に記載して、ストーリーマップの該当する位置に貼る

条件	内容
独立していること (Independent)	他のストーリーと独立していること
交渉可能なこと (Negotiable)	開発側とプロダクトオーナーの間でコストや開発方法について交渉により改善が行えること
価値があること (Valuable)	利用者にとって価値がある事
見積もりが可能なこと (Estimate)	コストが見積もれるだけの情報があること
小さいこと (Small)	短い期間で製造ができるような、適切なサイズに分割が行われていること
テスト可能なこと (Testable)	そのストーリーが完了したかテストできること そのテストにOK・NGと言えるための明確な受入条件があること

3) スプリント計画

プロダクトバックログの作成により、優先順位がつけられた、実装すべき要件の一覧が得られた。**次に、各要件を実際の開発が行える具体的な作業(タスク)レベルまで詳細化する。**

プロダクトバックログの中で優先順位が低いものは、その時点ではまだ要件にあいまい性があって、タスクレベルに落とし込めないものがある可能性がある。また、実際の**実装を行う「スプリント」は短い実装期間（1～4週間）となる。この期間内に実装が可能なものを、プロダクトバックログの中から範囲を決めて開発に着手する**必要がある。



3) スプリント計画

→ プロダクトバックログの中から **1回のスプリントでどこまで実装ができるのかという計画を立てる必要があります、この計画を「スプリント計画」と呼ぶ。**

スプリント計画では、「どのような価値のあるものを、どこまで作るのか」「達成するにはどうすればよいのか」を開発チーム内で話し合い、合意を行う。これにより、1回のスプリントで実装する範囲が決まり、この実装範囲にある要件をタスクレベルに落とし込んだタスク一覧を「スプリントバックログ」と呼ぶ。

ストーリー	見積り
実現したいこと#1	2
実現したいこと#2	1
実現したいこと#3	3
実現したいこと#4	5
実現したいこと#5	1
実現したいこと#6	3

.....

1回目のスプリント
で実現(見込み)

実現したいこと#1(〇〇ができるようになる)

タスク1) × × を作成する
タスク2) × × を作成する
タスク3) 処理結果を表示する

3)スプリント計画の流れ

流れ1)プロダクトオーナーが開発チームに対して、今回のスプリントでプロダクトバックログの中からどこまで実装するのかを説明し、最終的な達成目標はどこまでかについて合意する。

プロダクトオーナーは、今回のスプリントで最終的に何がほしいかをプロダクトバックログの中から実装順番を決めていく。また、実装方法の内容(具体的な画面イメージや動き、結果表示、エラー)も伝える。開発側は、今回のスプリントでどこまで作れるか、どうやって実現するかを決めていく。

流れ2)主に開発側の作業で、実装対象に必要なタスクをすべて洗い出す。プロダクトバックログの中の今回の実装範囲にあるストーリーを、実装に必要な具体的なタスクに落とし込む。場合によっては、ストーリーの実装が1回のスプリント内で収まらない可能性もある。この場合は、プロダクトオーナーと相談して、実装範囲を調整するなどする。

構成人員

プロダクトオーナー(1名):プロダクトの価値を最大化
プロダクトバックログの管理者
開発チームに相談できるが干渉できない
開発チーム(3~9人):ソフト開発を行う
肩書やサブチームは無し

3) スプリント計画の見積もり

アジャイルで用いられる見積もり方法として、「プランニングポーカー」を用いた技法がある。プランニングポーカーとは、「1」「2」「3」「5」「8」…と各数字が書かれたカードを開発メンバ全員が持ち、あるストーリーの見積もりがどれくらいの値になるのかをカードで出し合って決定していく手法である。見積もられた値により、「1回のスプリントで開発が行えるストーリーの範囲」を「見積もり値の合計値」で定める。

a) バックログの中から基準(=1)となるストーリーを決める

→見積もり時間が短く、比較的正確さが高いストーリーを選ぶと良い。

b) 他のストーリーについて、基準からの相対値をカードで決める

→この相対値を、プランニングポーカーを使って求める

- ①見積もりを行うストーリーの機能イメージをプロダクトオーナーが説明する
- ②開発メンバ全員が、ストーリー開発に必要な見積もり値をカードから選び、出し合う
- ③カードの値が全員一致:見積もり値が決定する

不一致:見積もり値を出した理由を聞き出し、全員が合意または再びカードを出し合って一致するまで行う

課題14-3:締切8/1

さきほど作成した進路決定のプロダクトバックログとスプリントバックログを作成し、ベロシティの見積もりを作成ください。隣の学生と共有して、意見交換ください。

ストーリー	見積り
実現したいこと#1	
実現したいこと#2	1
実現したいこと#3	

基準

ストーリー	見積り
実現したいこと#1	2
実現したいこと#2	1
実現したいこと#3	3

基準から2倍と想定

基準から3倍と想定

見積もられた工数を「ベロシティ」という

ストーリー	見積り
実現したいこと#1	2
実現したいこと#2	1
実現したいこと#3	3
実現したいこと#4	5
実現したいこと#5	1
実現したいこと#6	3

.....

1回目のスプリント
で実現(実績)

2回目のスプリント
で実現(見込み)

3回目のスプリント
で実現(見込み)

1回のスプリントで行えるベロシティが6とすると

4) スプリント

スプリント計画で合意したストーリーについて、設計・製造・テストを行う。この期間は、主に1～4週間となる。

製造時においては、実装しなければならないすべてのタスクをタスクボードに貼り付け、あるタスクが「実装前」「実装中」「実装完了」のどれにあたるかを開発者全員が把握できるようにする。

スプリント中の活動では、「プロダクトバックログ」、スプリントで必要な作業のリストである「スプリントバックログ」、働くソフトウェアである「インクリメント」の3つが作成・更新される。

5) デイリースクラム

デイリースクラムは、開発者全員で行う情報共有のための簡単なミーティングである。主に、作業開始前の朝などに実施します。このミーティングを通して、各メンバに於いての作業状況や課題点などを全員が把握し、1日の作業内容を決めていく。

アジャイル開発のチームは、自己組織化したチームであるため、やるべき作業やメンバが困っている作業に対して、誰もが積極的に手を挙げてサポートする。デイリースクラムは、メンバの状況を共有しあい、お互いにサポートできることを確認しあう場となる。例えば、以下のような質問に対する回答を話し合う。

- ① 昨日行ったことは？
- ② 本で行うことは？
- ③ 現在の課題は？

Trello



<https://www.renjusha.net/renju/leaflet/nyuumon.pdf>

6) スプリントレビュー

スプリントで作成された成果物を発注者側と確認し、実装を計画したストーリーについての完了判断を行う。このレビューを「スプリントレビュー」と呼ぶ。成果物は実際に動作するものであるため、このレビューにより改善点などが見えるようになる。

スプリントレビューでは、プロダクトオーナーや関係者に対して成果物のデモを行い、スプリントで予定していたストーリーがどこまで完了したか、完了したストーリーが適切にできているか、そのストーリーがビジネスにおいて価値あるものになっているかなどを確認する。そして、プロダクトオーナーや関係者は、確認したストーリーの実装に対してOK(Done)を出すか、NGを出すかの判断をする。

- ・ Doneの定義

各ストーリーに対する明確なゴール(条件:Doneの定義)を定める必要がある。

7)ふりかえり(レトロスペクティブ)

開発者全員で、今回のスプリントについて、よかったことや課題の確認を行い、全体の見直しをはかる。

ふりかえりの方法はいくつかの手法がありますが、主に「よかったこと」「問題点」「次に行うべきこと」の3つを書き出して行う方法などがある。

