
Less



12-14 October

https://less.works/course-details/certified-less-practitioner-tokyo-4457



健全なチームの
動きを
密かに計測する
方法

Secret
measurement
of healthy team
dynamics



Founding Wärtsilä

Wärtsilä was founded in 1834 in Tohmajärvi, Finland, starting as a sawmill and ironworks company.

Global leader

in decarbonising marine and energy

Founded in

1834

Net sales, MEUR

5,842

Our personnel

17,581

Nationalities

127

Country presence

79

Locations worldwide

242

Figures from 2022

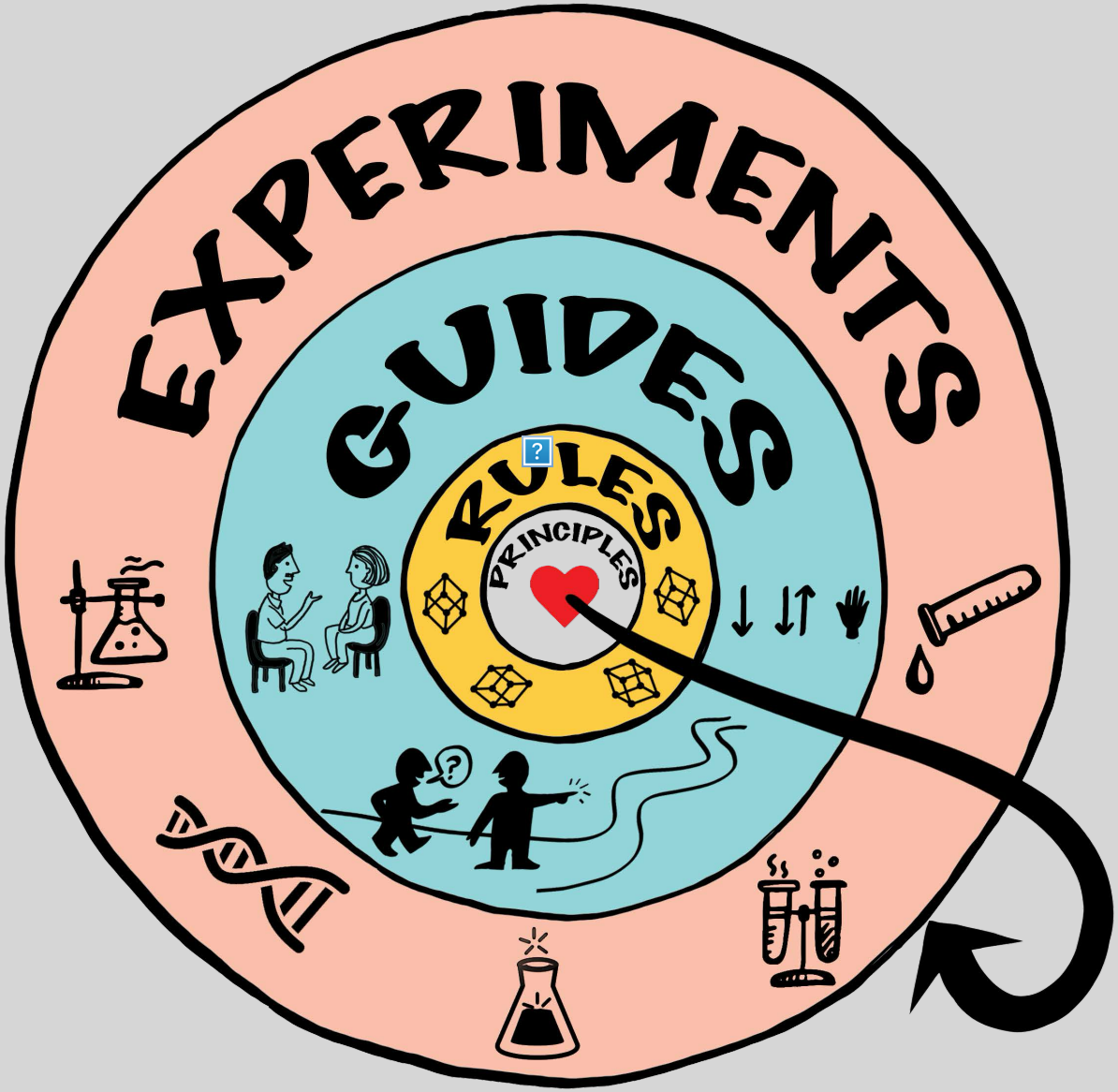






©

Large Scale Scrum



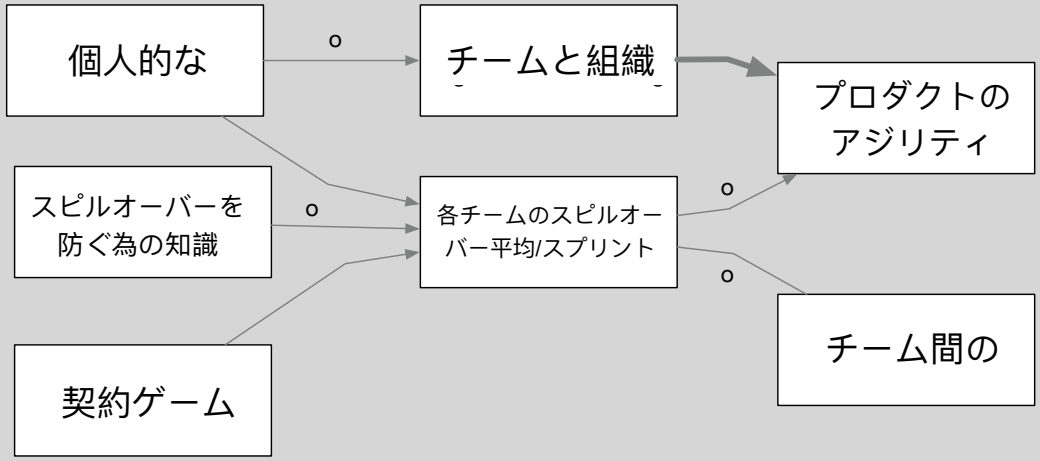
http://less.works BY-ND

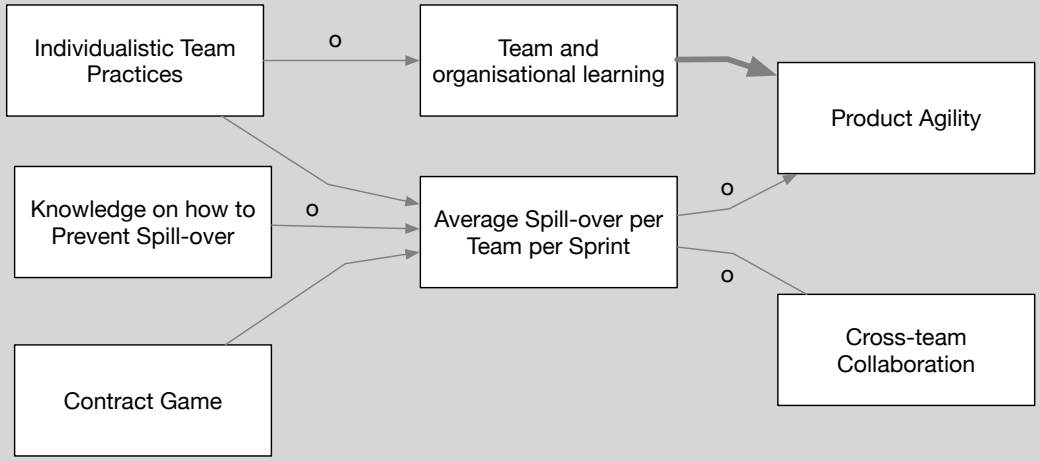


なぜそんなに
スピルオーバーに執着
しているの?
Why are you always so
obsessed with spill-over?

アジェンダ

Agenda





スピルオーバーとは?

What is spill-over?





スピルオーバーの定義

スプリントの終わりに着手中となっているアイテム数

Definition Spill-over

Amount of items that are in progress at the end of the Sprint



例

	選択アイテム数	スプリント中に着手されたアイテム数	Doneしたアイテム数	スピルオーバー
例 1	10	5	4	1 (5じゃない!)
例 2	1000	3	3	0
例 3	10	10	6	4

Examples

	Items Selected	Items Started During Sprint	Items Done	Spill-over
Example 1	10	5	4	1 (not 5!)
Example 2	1000	3	3	0
Example 3	10	10	6	4



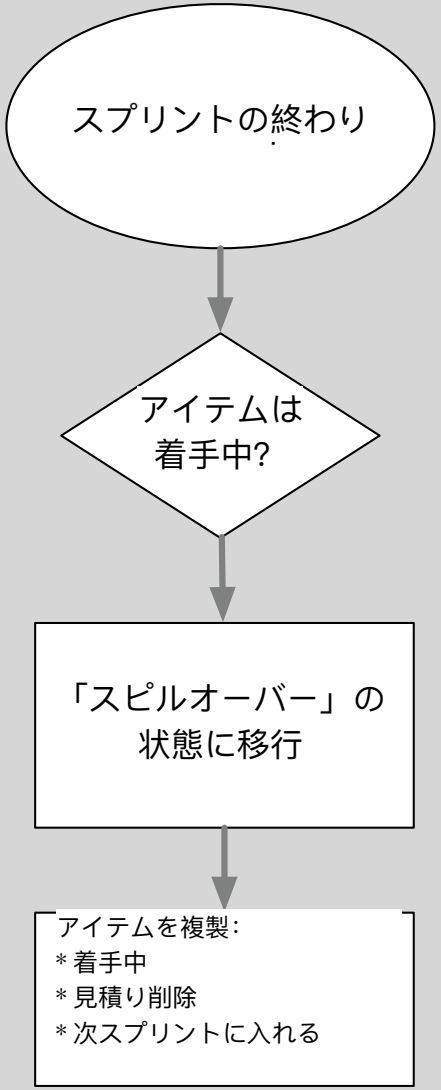
予測の為ではない!

say-do指標や計画の精度とは一切関係ない

NOT ABOUT　PREDICTABILITY!

Not related to say-do metric or planning accuracy in any way

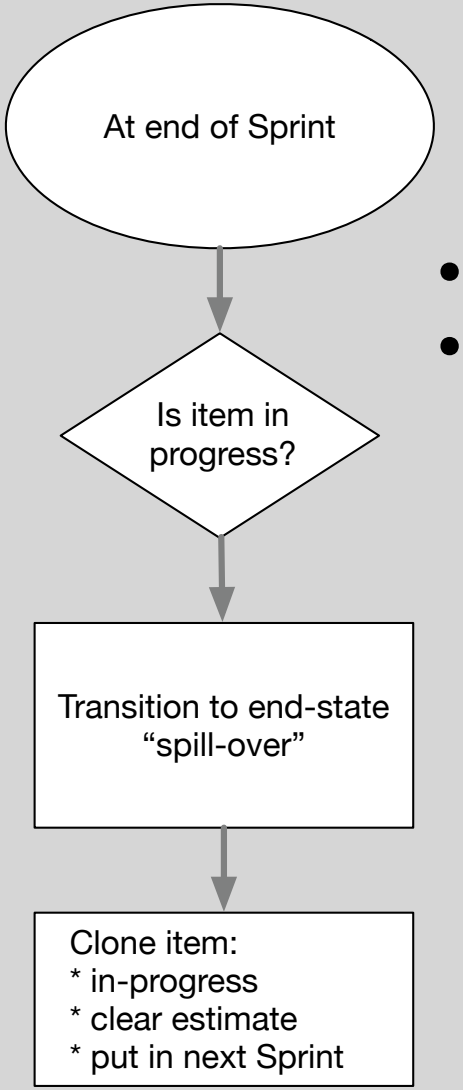
計測が簡単



- スピルオーバーは終了事の状態
- スピルオーバー指標:

  スプリント単位にスピルオーバーを数える

Easy to measure



- Spill-over is end-state
- Spillover metric:

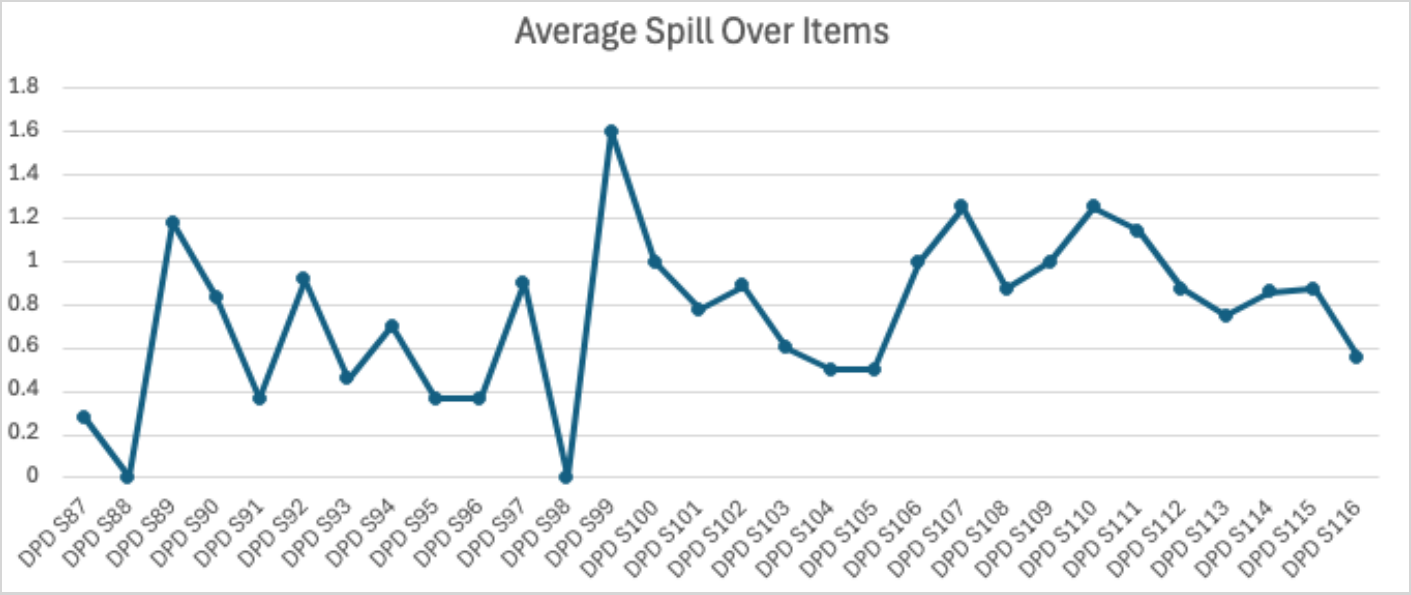
  for each item in Sprint x
  count spillover state

LeSS環境でスピルオーバーを計測

Measure Spill-over in LeSS

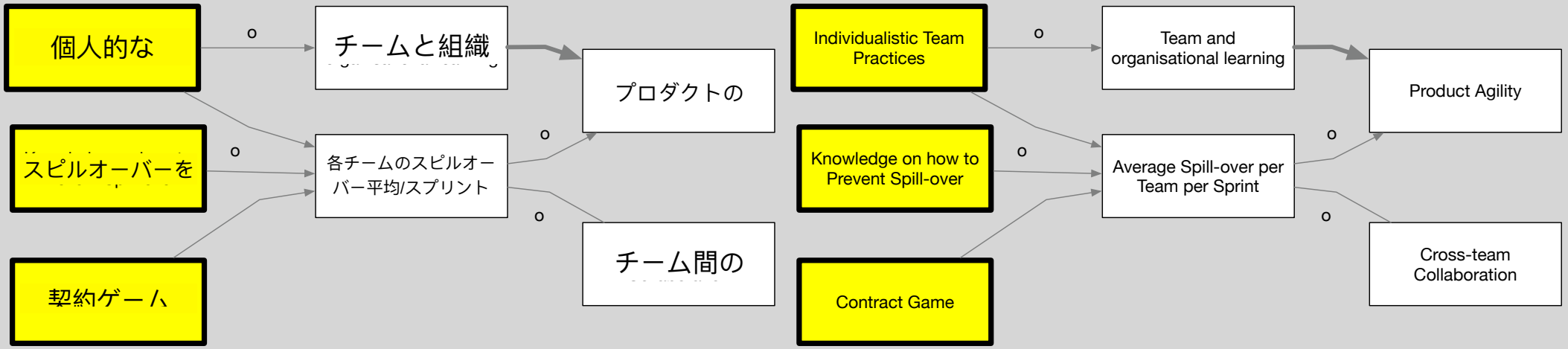
プロダクトのスピルオーバー合計 / チーム数

Total spill-over Product / amount of teams



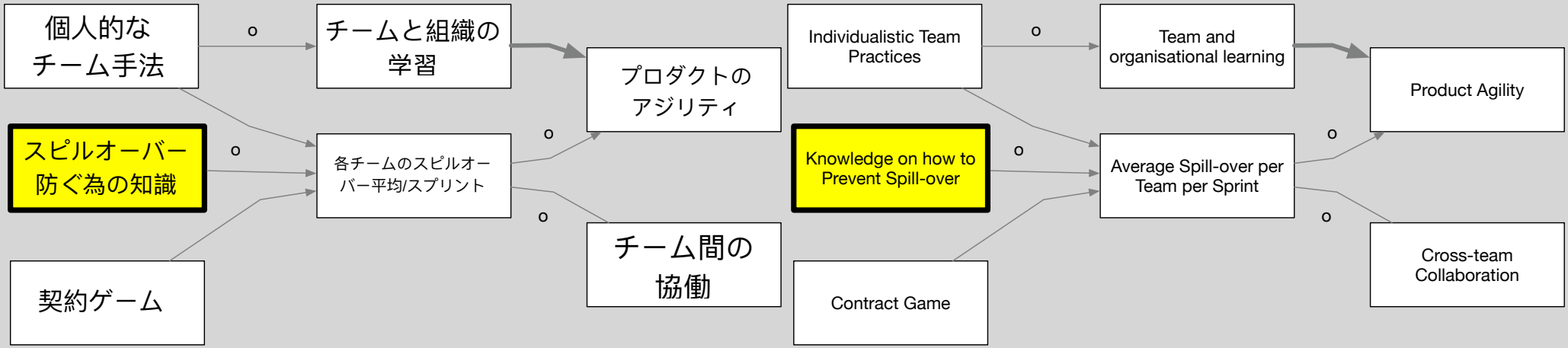
スピルオーバーの原因は?

What causes spill-over?



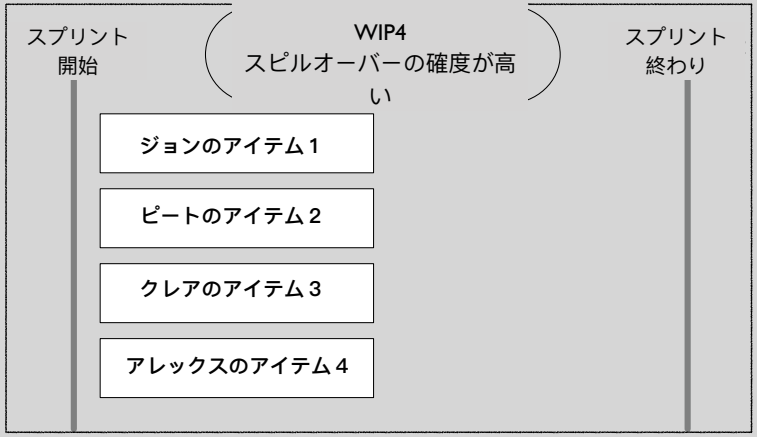
スピルオーバーは
防げる!
(ほとんどの場合)

Spill-over can be
prevented!
(most of the time)

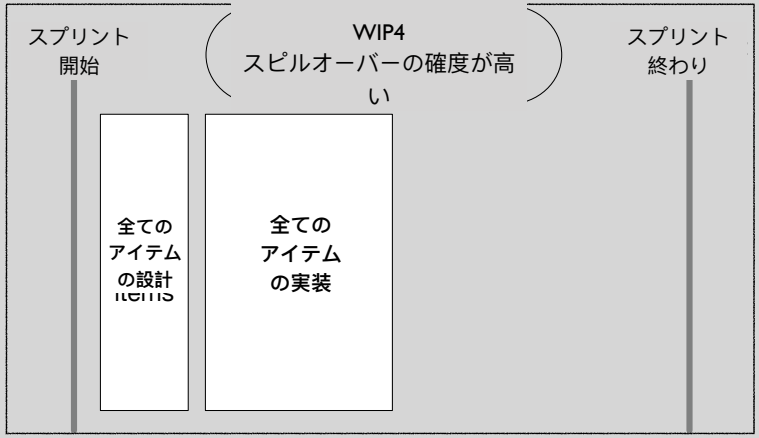


予防法その 1:WIPを減らす

高WIPの例



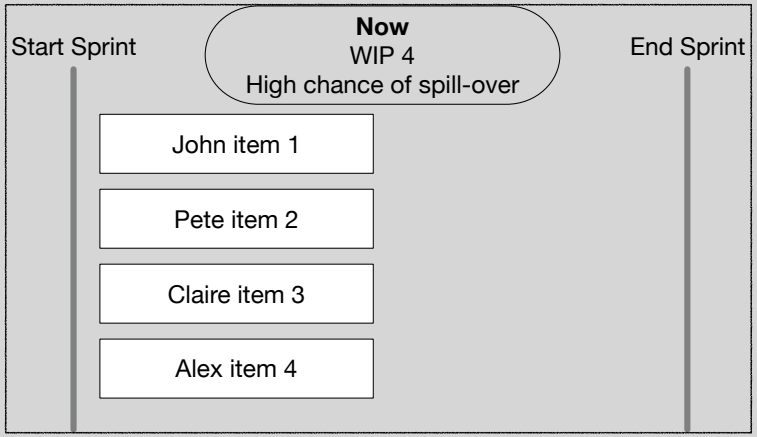
例：各メンバーがそれぞれアイテムを所有



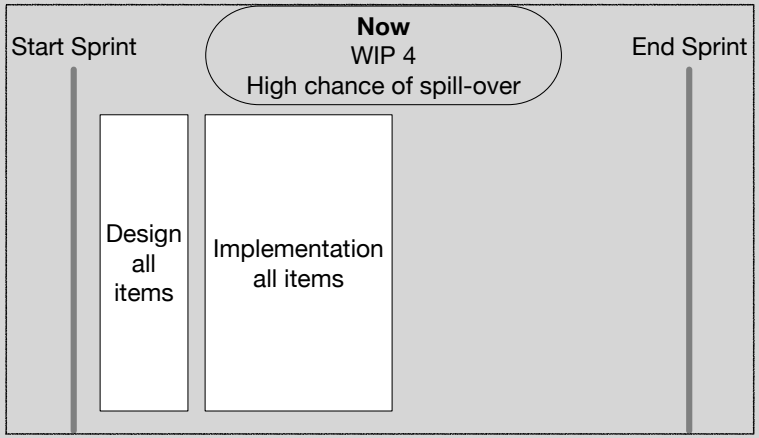
例 2： ウォータースクラム

Prevention tactic 1:Reduce WIP

High WIP Examples



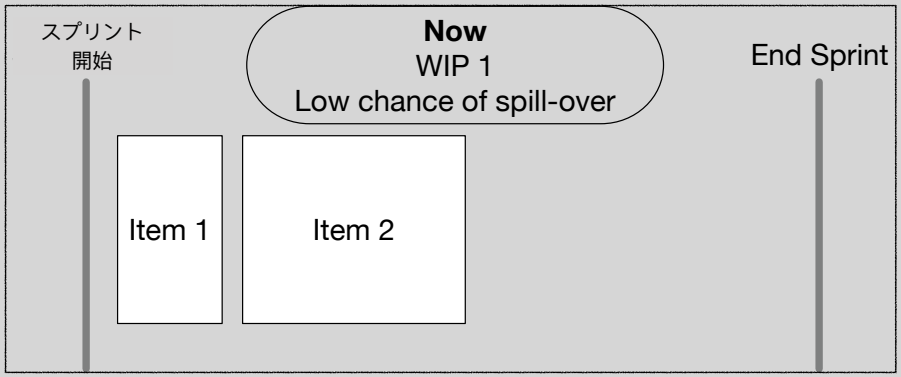
Example 1: Each member has their own items



Example 2:WaterScrum

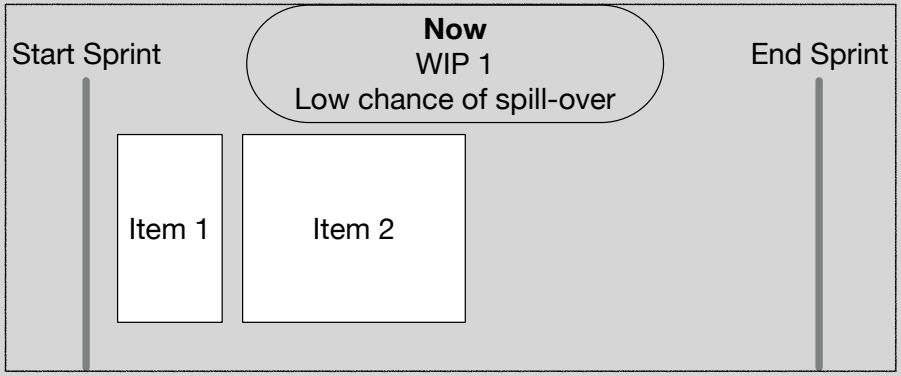
予防法その 1:WIPを減らす

低WIP
の例



Prevention tactic 1:Reduce WIP

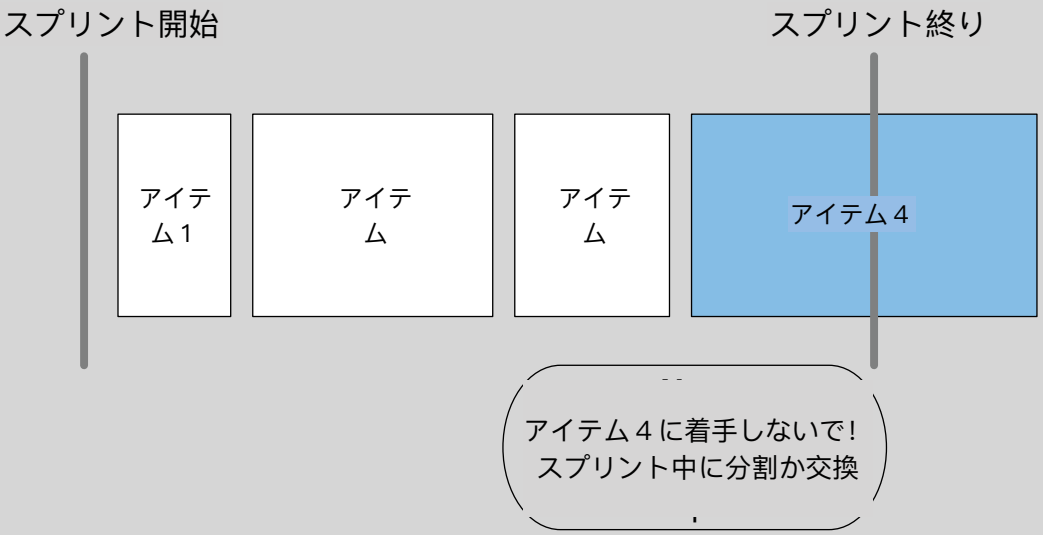
Low
WIP
Examples

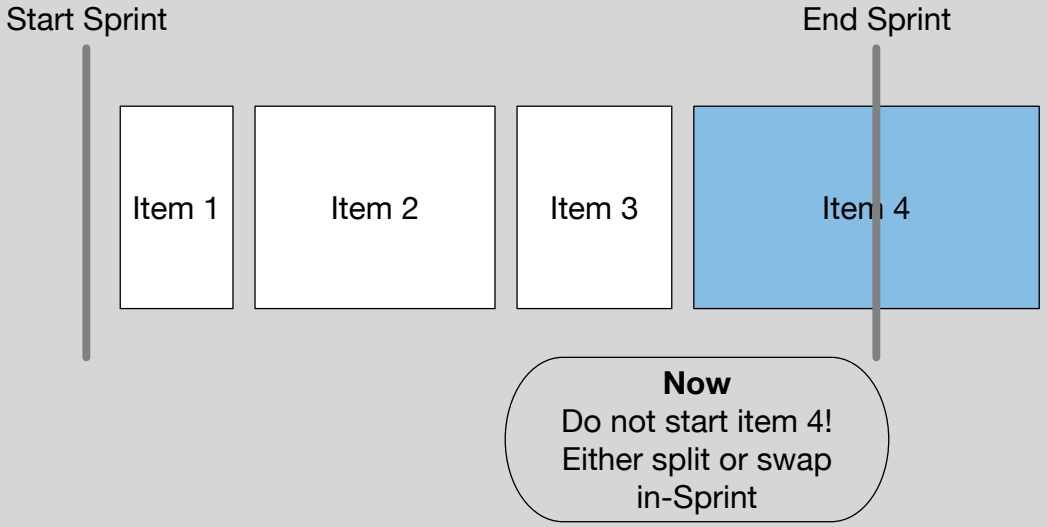


Example 3: One item at the time

予防法その2:
分割か交換

Prevention tactic 2:
Split or Swap

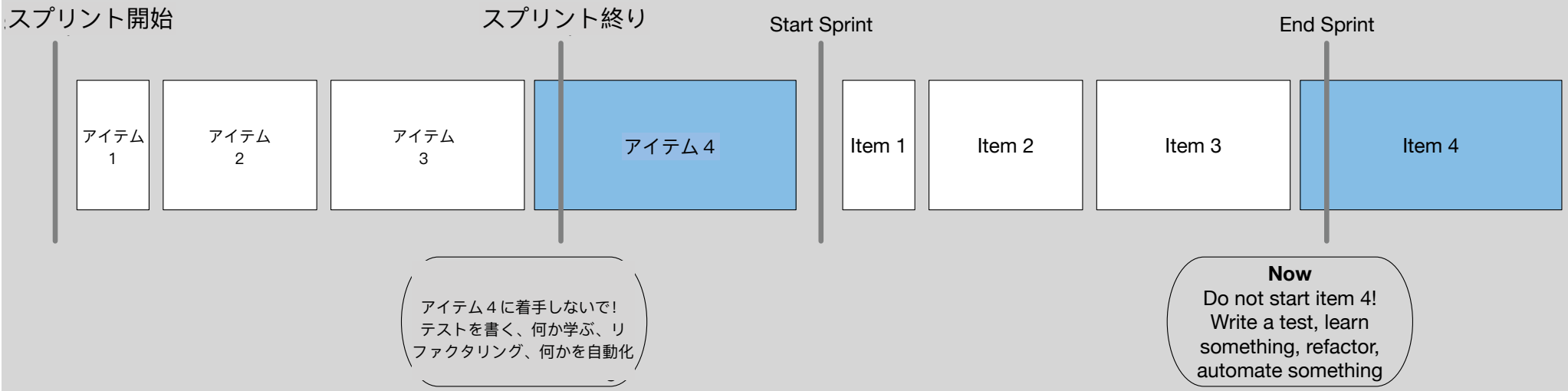






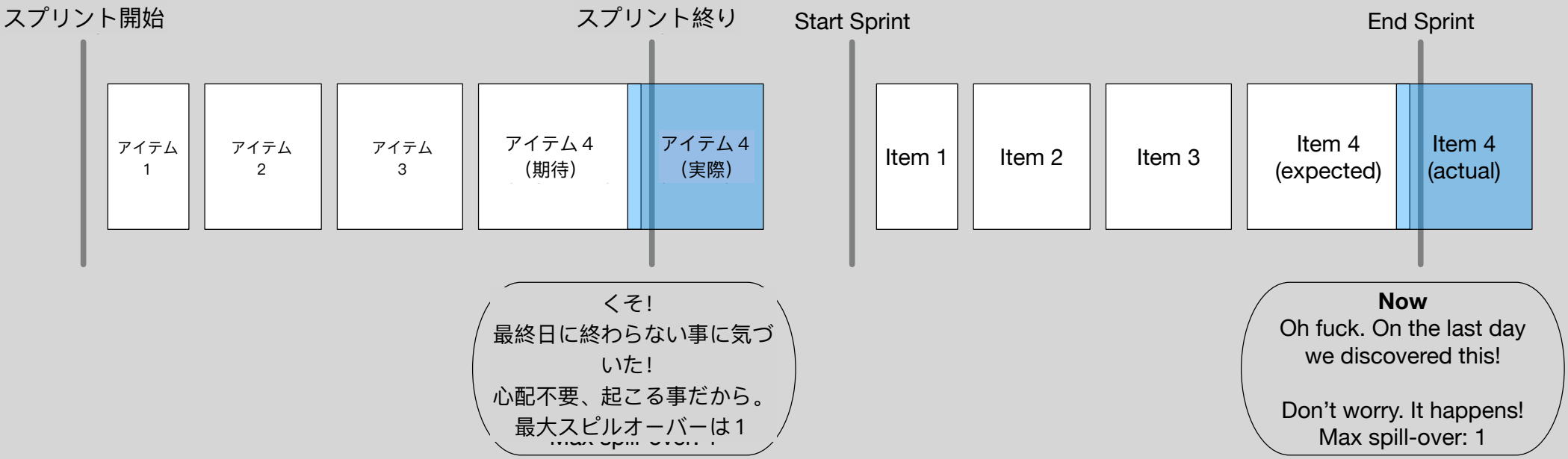
予防法その3: 最終日に着手しない!

Prevention tactic 3: Don’t start last day!



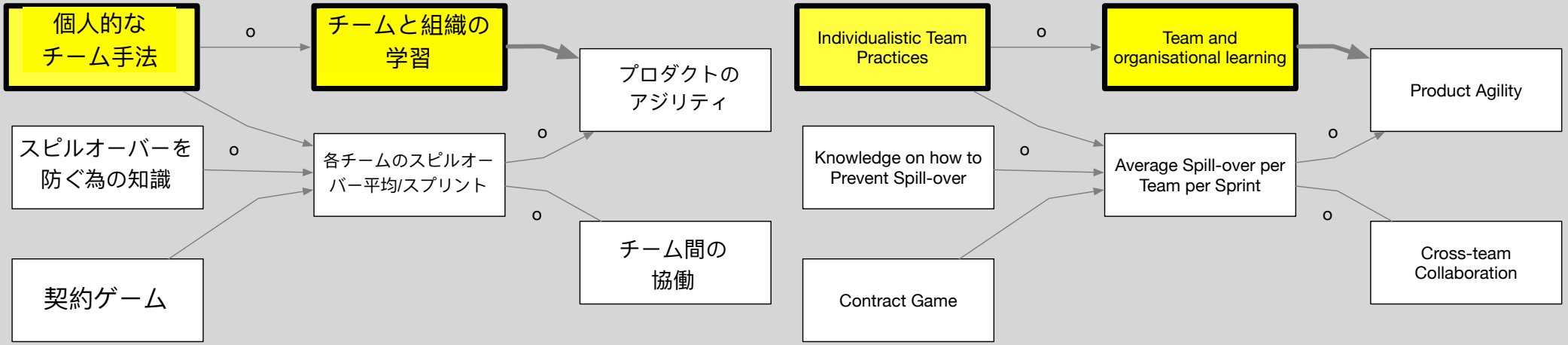
その 4:
くそ!

Tactic 4:
Oh Fuck!



チームダイナミクス!

Team dynamics!



less.worksのブログ参照: チーム内とチーム横断での協働手法

See less.works blog: In-team and cross-team collaboration practices



避ける: 個人的なチーム手法

- 各メンバーがそれぞれ別アイテムを担当
- 機能単位に専門化されたシーケンシャルなプロセス
- メンバーが自分にスキルがある仕事しかしない
- フィーチャーホギング又の名を「スプリントって何? 」
- 依存関係によるブロッキング

Avoid: Individualistic internal team practices

- Each member has their own items
- Sequential process with functional specialization
- Each member sticks to their single skill
- Feature hogging, also known as “What Sprints?”
- Blocked by dependencies

これらのプラクティスはいずれも、**WIPの増加、協働の減少、知識共有の減少**につながる

Each practice leads to **high WIP** and **low collaboration** and **low knowledge sharing**



試して: 協働的なチーム内手法

- スプリントプランニング2での設計と、細かい粒度のタスク
- 1日に複数回のデイリー
- 複数のペア
- モブプログラミング
- スウォーミング
- チームメンバーを一人犠牲にする

Try: Collaborative internal team practices

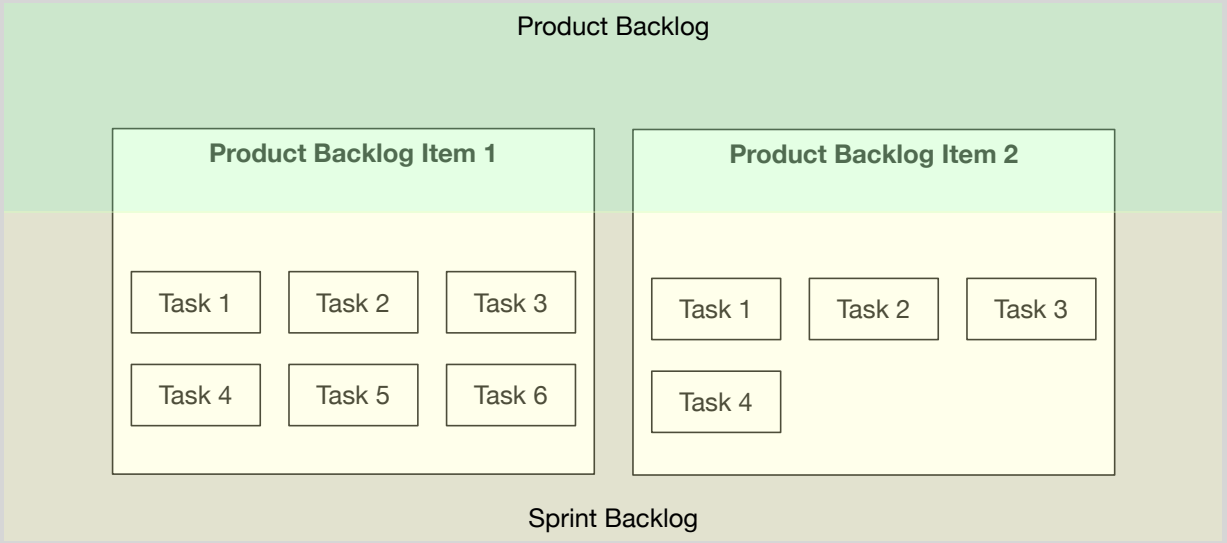
- Design in Sprint Planning 2 and fine-grained tasks
- Multiple daily
- Multiple pairs
- Mob programming
- Swarming
- Sacrifice a team member

Each practice leads to **low WIP** and **high collaboration** and **high knowledge sharing**

Each practice leads to **low WIP** and **high collaboration** and **high knowledge sharing**

WIPとは?

What WIP?

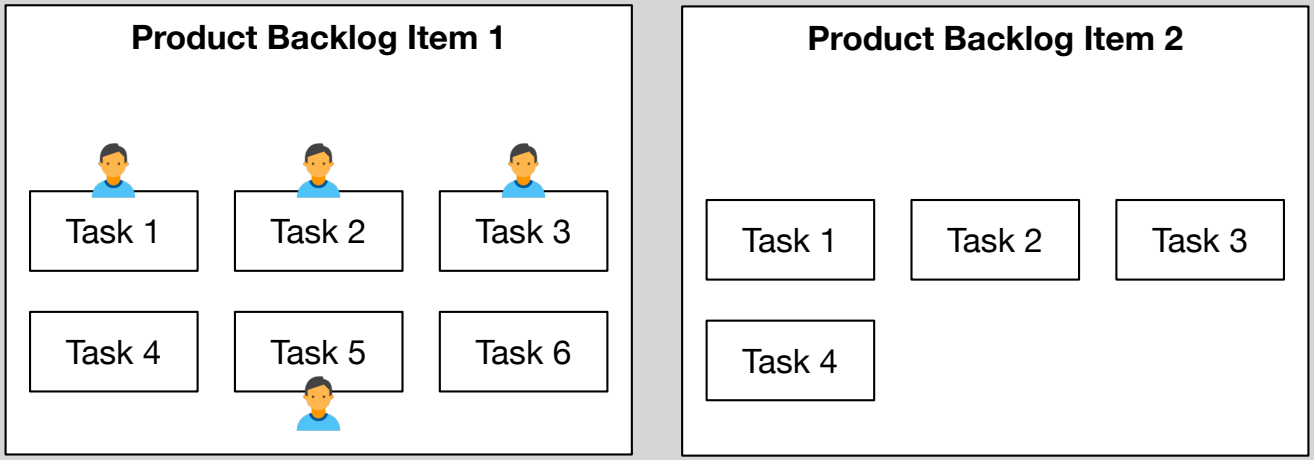


私たちが最も重視するのは、プロダクトバックログレベルのWIPである。なぜなら組織のアジリティに影響するからだ

We care most about WIP on Product Backlog level as that affects organisational agility

スプリントプランニング2での設計と、細かい粒度のタスク

Design in Sprint Planning 2 and fine-grained tasks



- スプリントプランニング2で設計することで、全員が同じ文脈を共有できる
- 細かい粒度のタスクにより、作業を分割できるようになる
- **1日に複数回**のデイリーにより、同期の機会が増える

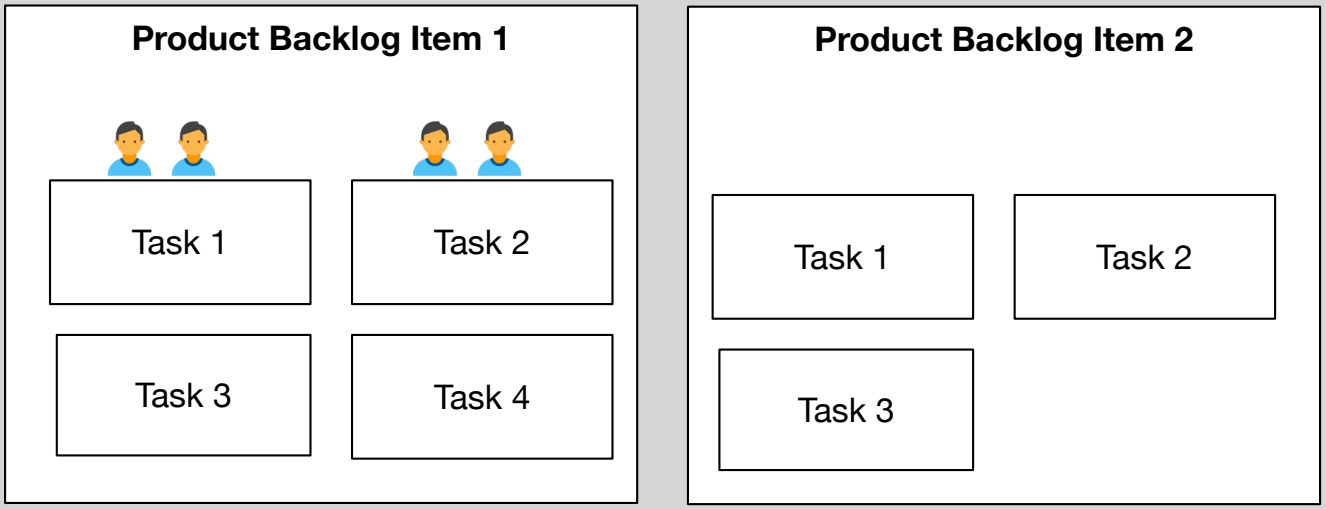
- Design in Sprint Planning 2 gives everyone the same shared context
- Fine grain tasks give the ability to split work
- **Multiple daily** give additional sync points


INSTALLATION
CONFIGURATOR
Default
Starting Pol
About Diego
BE
FE
zero sugar



複数ペア

Multiple Pairs



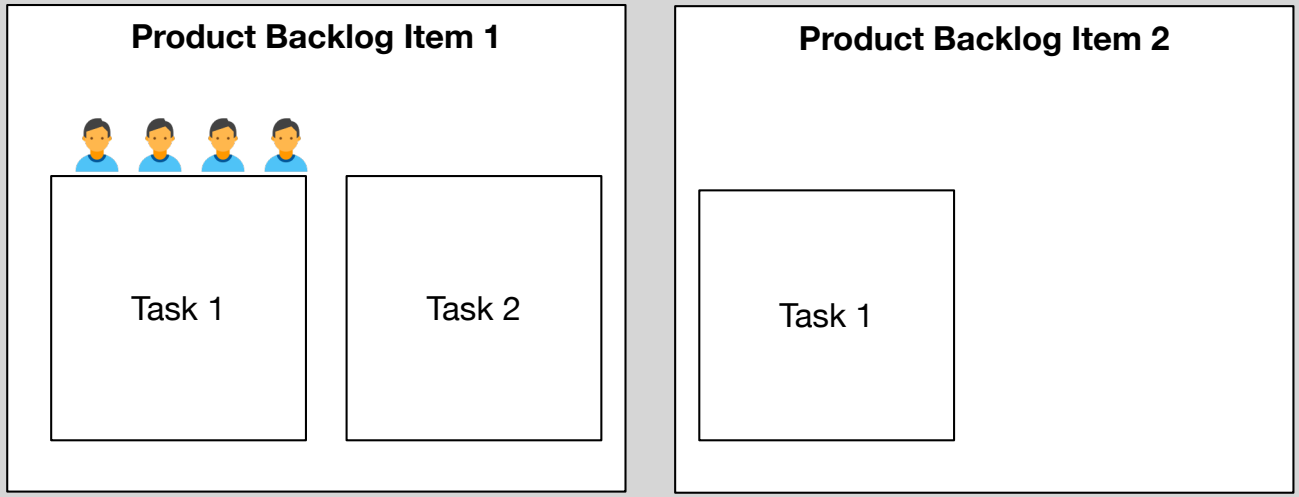
- 同期の必要性が減る
- タスクをより大きくできることが多い

- Reduces synchronisation needs
- Tasks can often be larger


weber

モブ (全員)

Mobbing (ensamble)



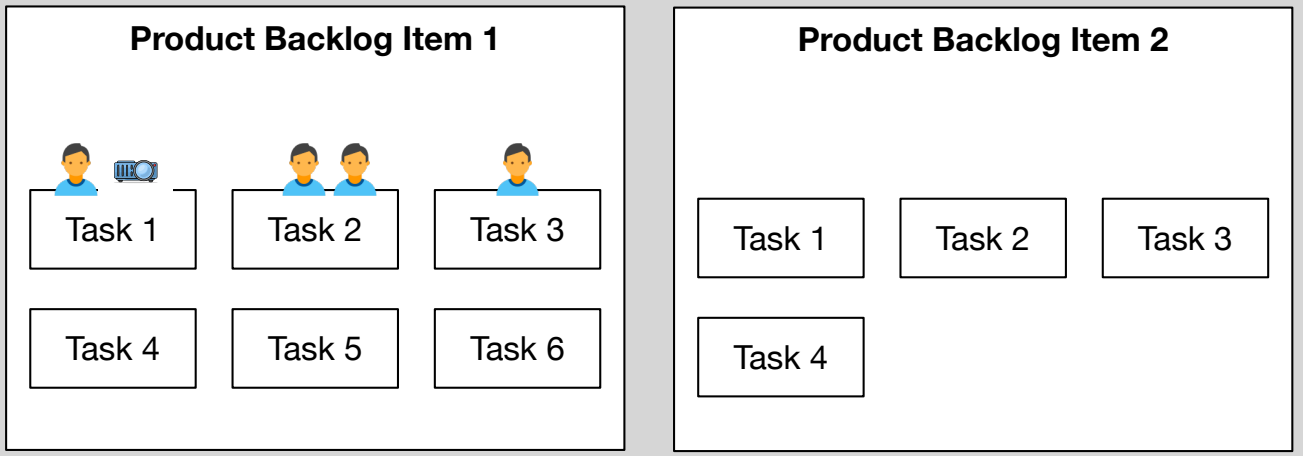
- アイテムとタスクのWIPが1（！）
- スプリントプランニング2が必要ないことが多い

- Item and task WIP 1 (!)
- Often not much Sprint Planning 2 needed



スウォーミング

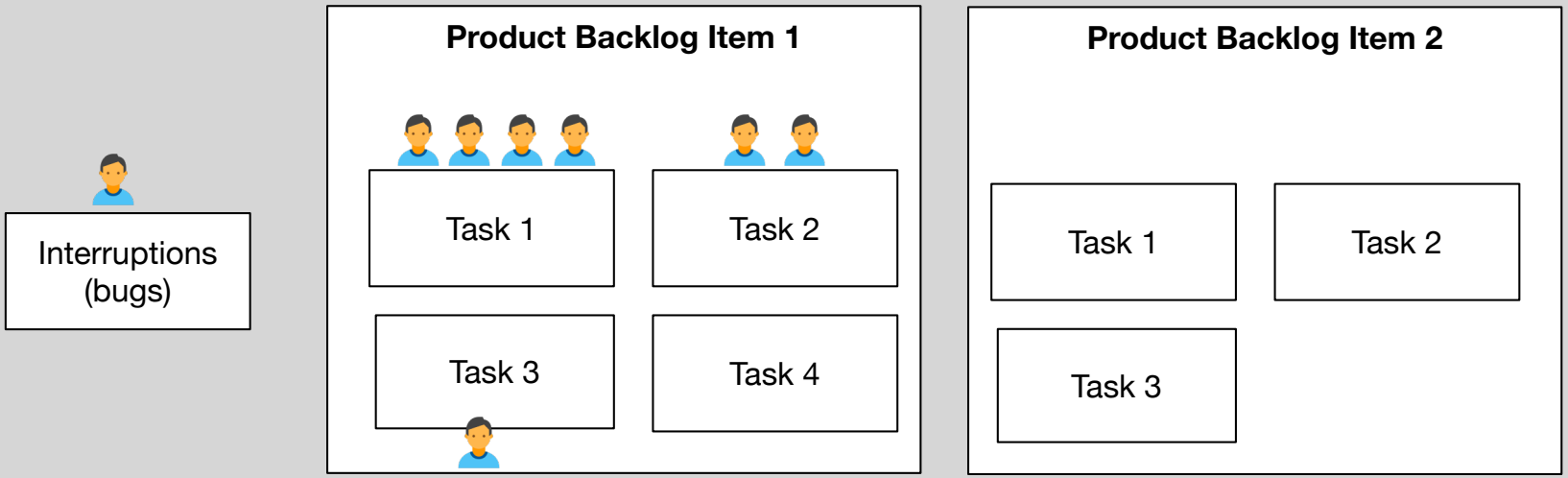
Swarming



- チーム全員が同じ場所にいる
- １箇所、全員が同期と全体把握できる場所がある
- 個人やペアがそれぞれ別のタスクに取り組む

- Whole team in same space
- One central point for synchronisation and overview
- Individuals and pair on separate tasks

組合せる (!)　Combining (!)

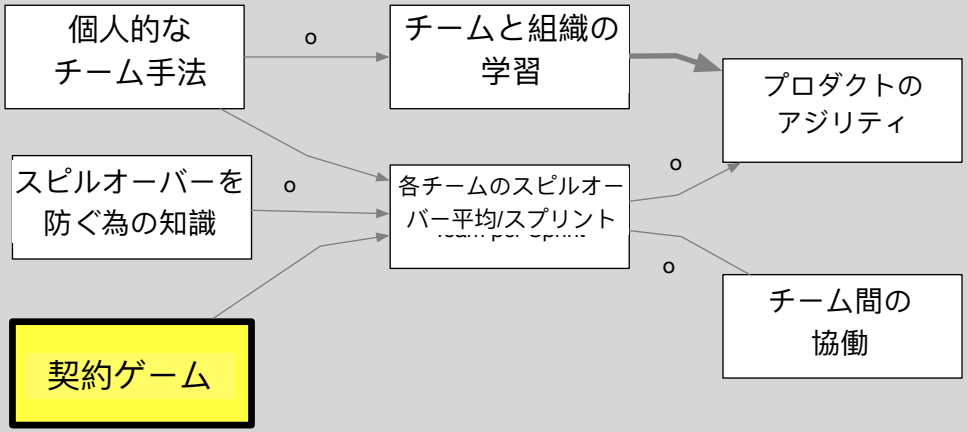


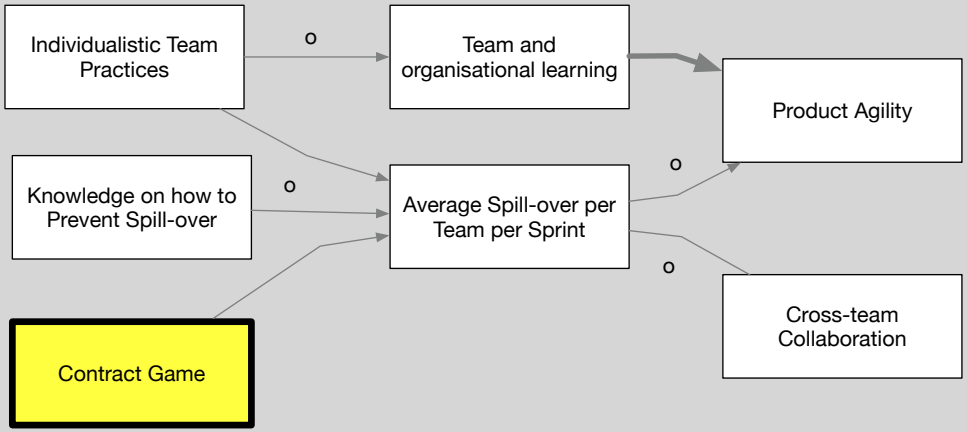
- 多くの場合、チームはいくつかの手法を組合せて使う
- **チームメンバーを一人犠牲**にすることで、チーム全体への割り込みを防げる

- Teams often combine different practices
- **Sacrifice a team member** can prevent interruptions to the whole team

契約ゲーム

Contract Game









プロダクト
マネージメント



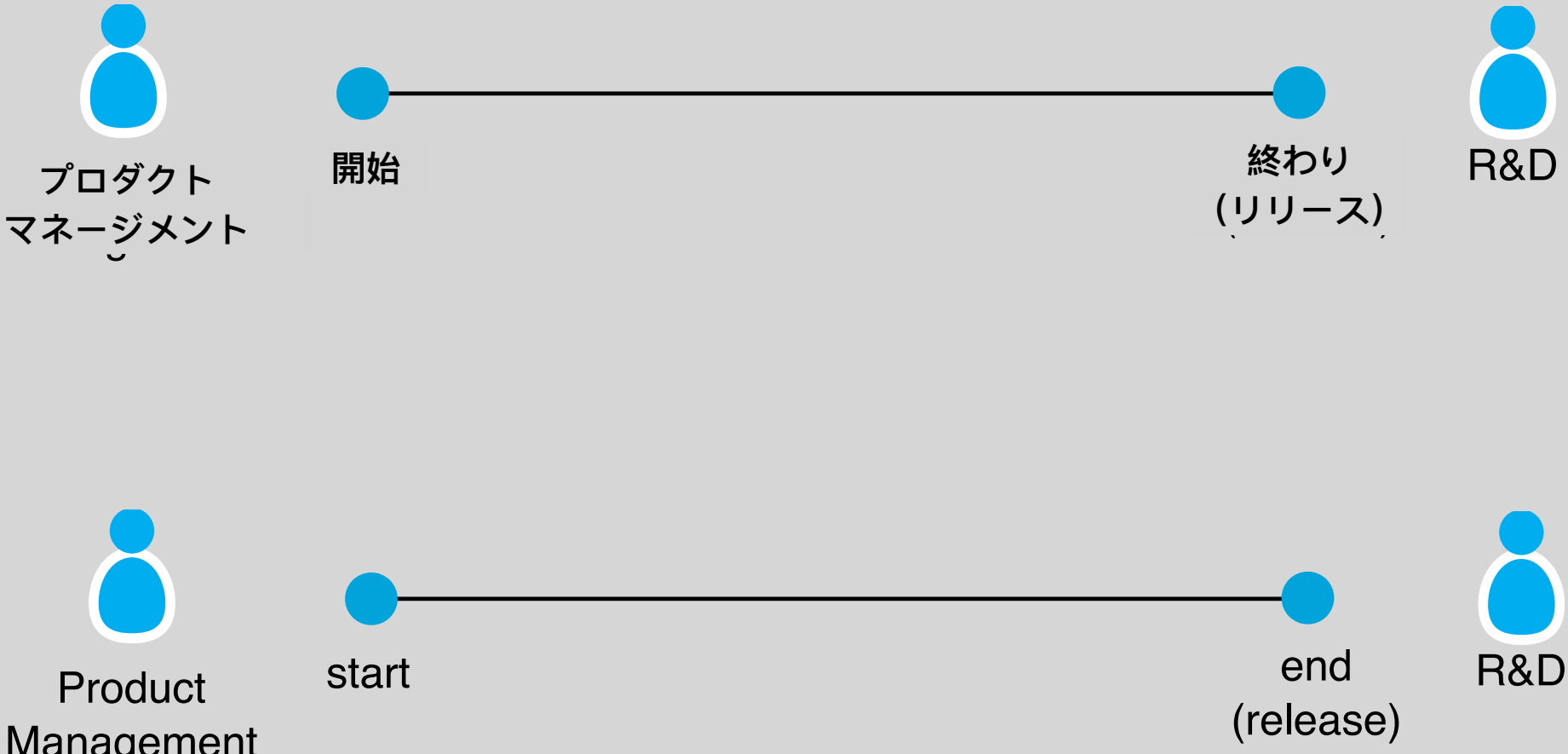
R&D

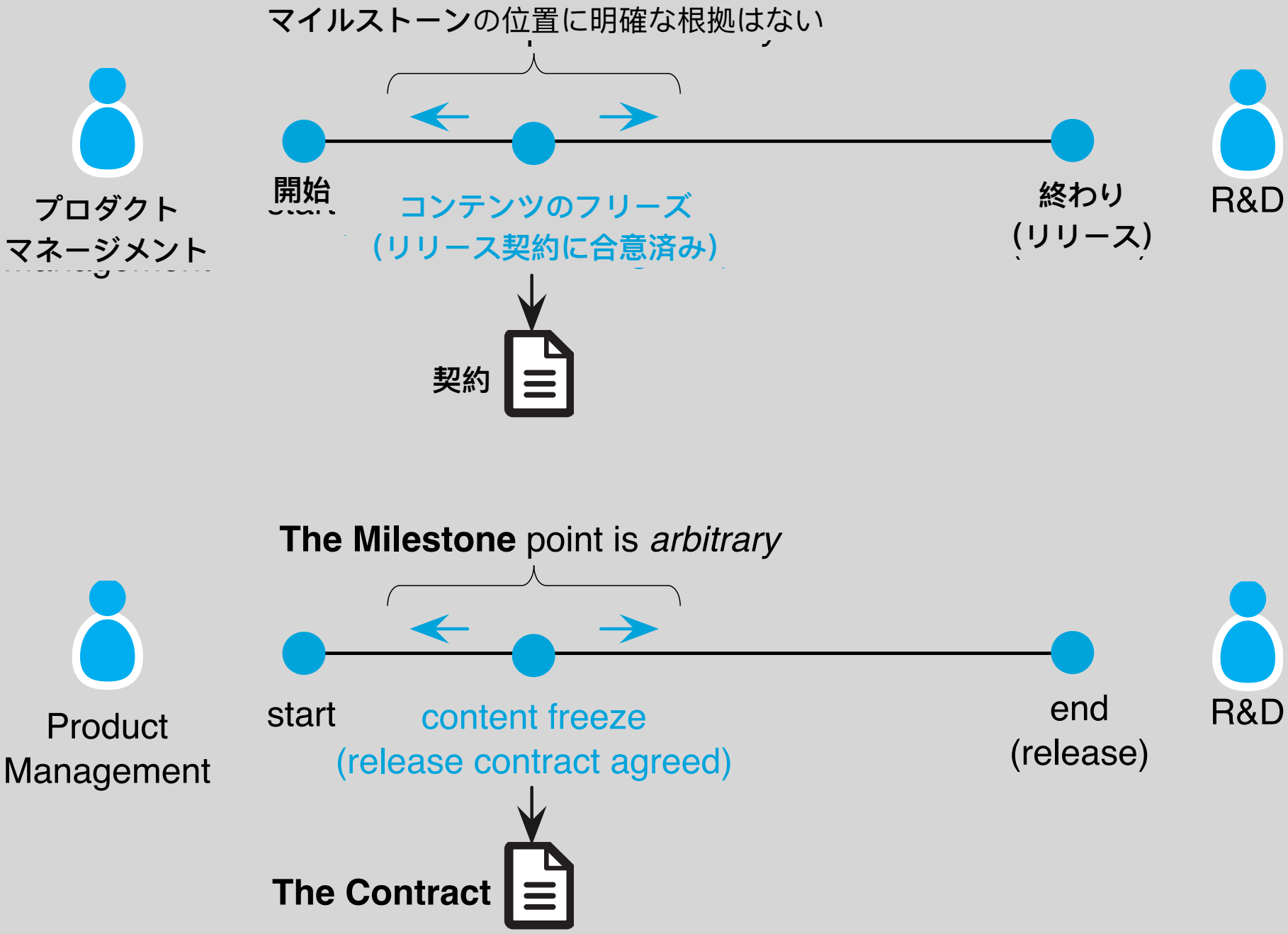


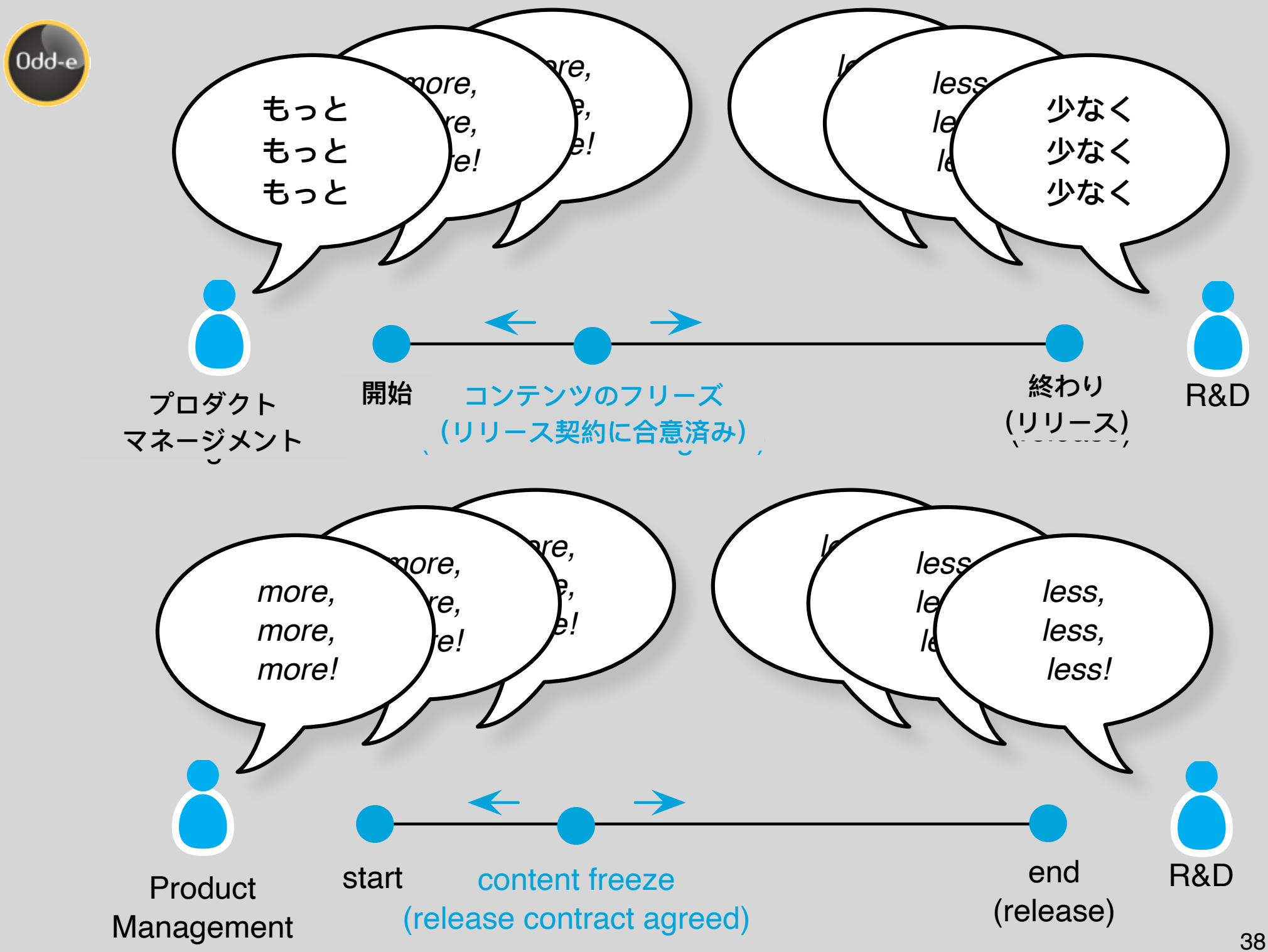
Product
Management

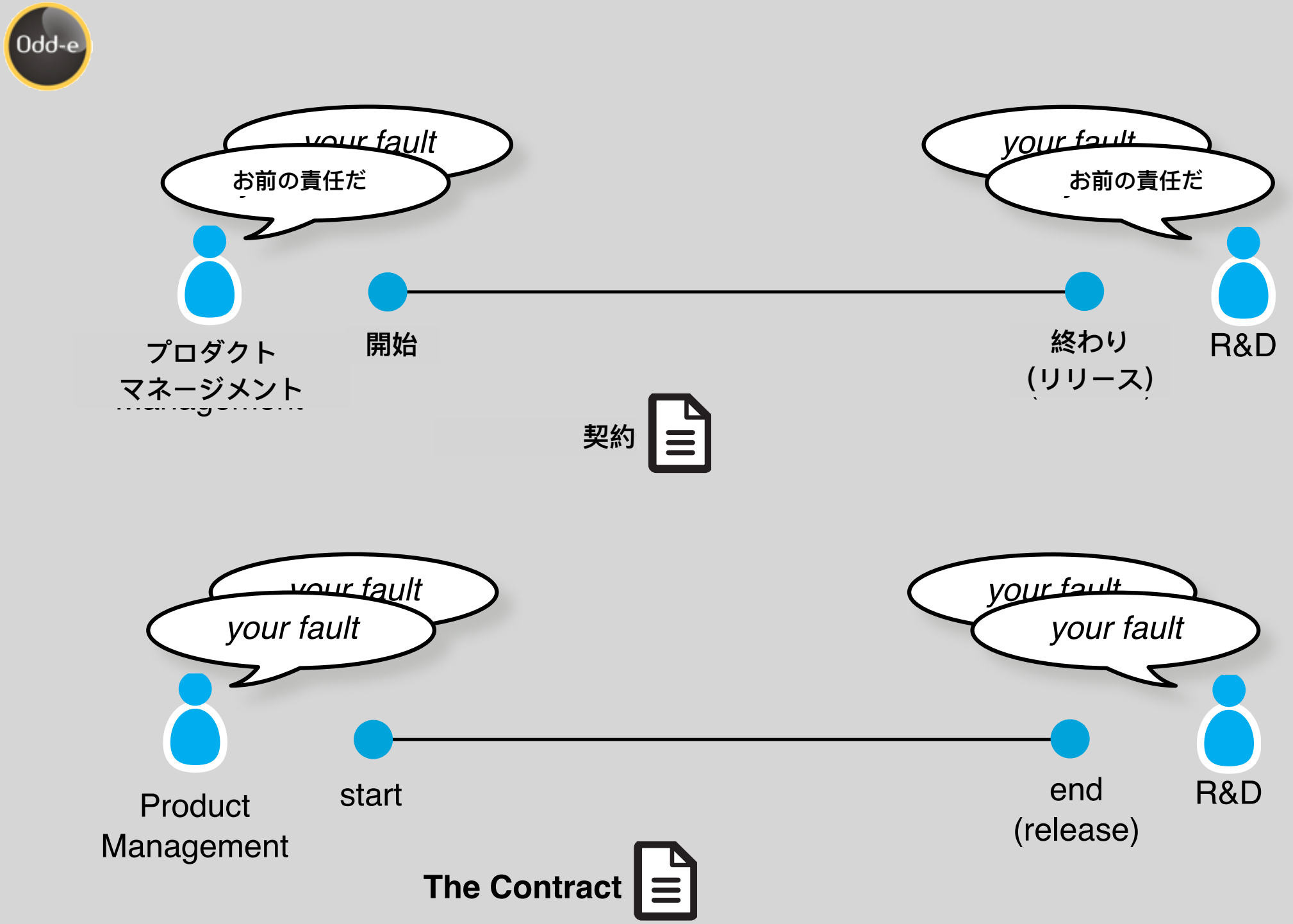


R&D


プロダクト
マネージメント
開始
終わり
(リリース)
R&D
Product
Management
start
end
(release)
R&D


マイルストーンの位置に明確な根拠はない
プロダクト
マネージメント
開始
コンテンツのフリーズ
（リリース契約に合意済み）
終わり
(リリース)
R&D
契約
The Milestone point is arbitrary
Product
Management
start
content freeze
(release contract agreed)
end
(release)
R&D
The Contract


もっと
もっと
もっと
少なく
少なく
少なく
プロダクト
マネージメント
開始
コンテンツのフリーズ
（リリース契約に合意済み）
終わり
(リリース)
R&D
more,
more,
more!
less,
less,
less!
Product
Management
start
content freeze
(release contract agreed)
end
(release)
R&D


your fault
お前の責任だ
your fault
お前の責任だ
プロダクト
マネージメント
開始
終わり
(リリース)
R&D
契約
your fault
your fault
your fault
your fault
Product
Management
start
end
(release)
R&D
The Contract

期限 / 約束

Deadlines / Promises

なぜ期限や約束はあるの?		Why Deadlines and Promises?	
厳しい外部制約 (例：規制当局、サポート終了)		Hard external constraints (e.g. regulator or end-of-life)	
同期 (例：マーケティングキャンペーン、プロセス変更)		Synchronisation (e.g. marketing campaign or process change)	
顧客への約束 (例：契約済みの契約に含まれている)		Promised to customer (e.g part of a deal that was sold)	
動機 (たいていは根拠のない期限)		Motivate (usually random deadline)	

期限 / 約束

なぜ期限や約束はあるの?	なぜ期限は変わるのか?
厳しい外部制約 (例：規制当局、サポート終了)	制約を満たしながらスコープ変更、または一時的な回避策
同期 (例：マーケティングキャンペーン、プロセス変更)	同期先の遅延
顧客への約束 (例：契約済みの契約に含まれている)	既存の約束よりインパクトのある新たな機会
動機 (たいていは根拠のない期限)	開発中に問題を発見

Deadlines / Promises

Why Deadlines and Promises?	Why deadlines change?
Hard external constraints (e.g. regulator or end-of-life)	Different scope for meeting constraint or temporary workarounds
Synchronisation (e.g. marketing campaign or process change)	Delay in dependency
Promised to customer (e.g part of a deal that was sold)	New opportunity with higher impact than previous promises
Motivate (usually random deadline)	Surprises discovered in development

計画駆動の

No plan-driven predictability

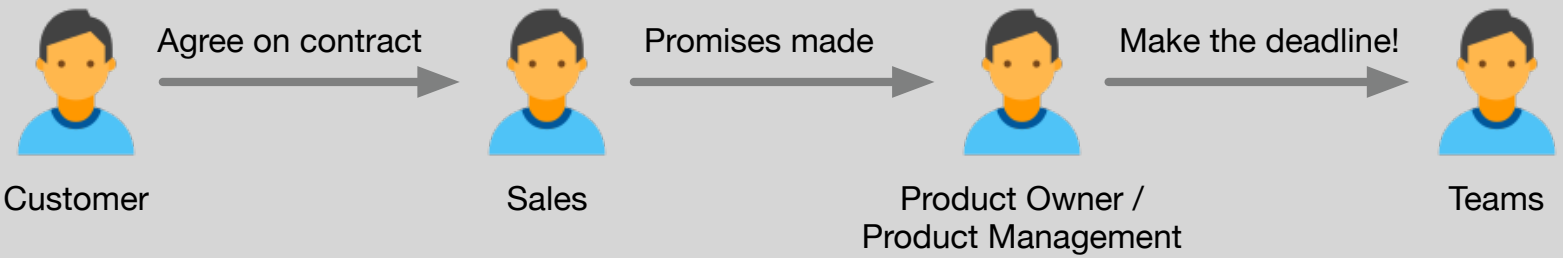
変化の激しい環境で、計画どおりに進めることにより、予測可能性を得ようとすると、硬直化を招き、プロダクトも人も壊してしまうような非現実的なコミットメントにつながる。

Predictability through following plan in dynamic environment leads to rigidity and unrealistic product- and people-destroying commitments

継続的な優先順位の見直しにより、計画ではなく実際のアウトカムを予測できる

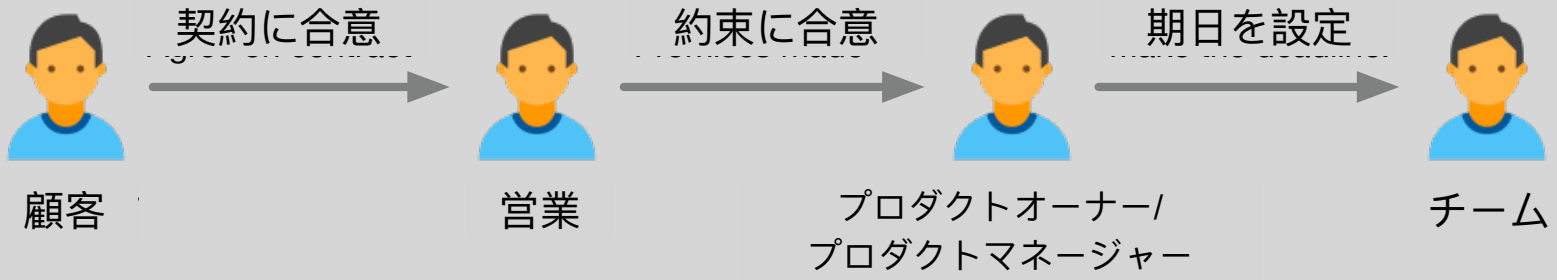
Predictability through continuous reprioritisation leads to not being predicable in plan but in outcome

Dealing with Deadlines



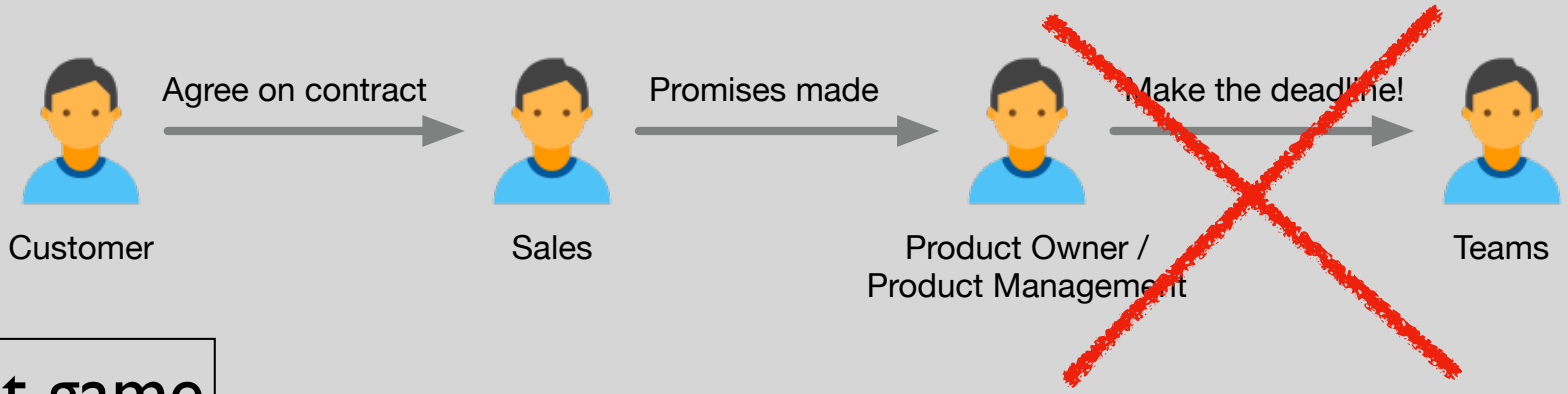
Contract game

期日との付き合い方



契約ゲーム

Dealing with Deadlines



Contract game

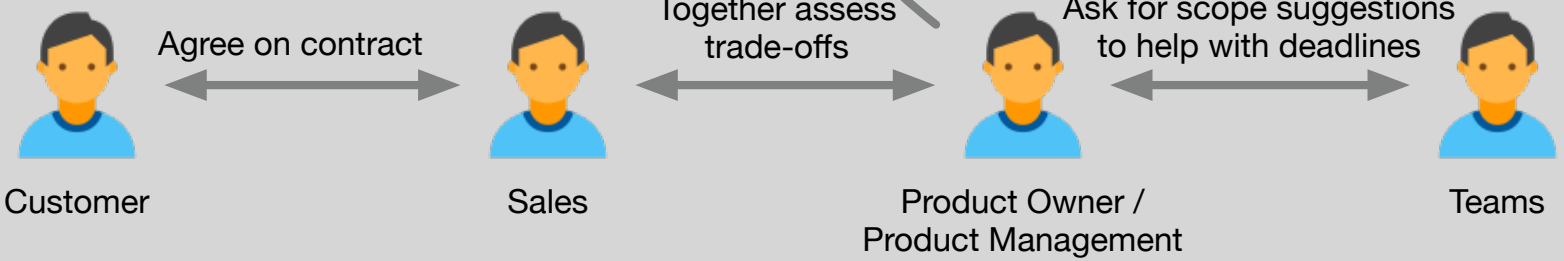
Collaborative development

Update priority

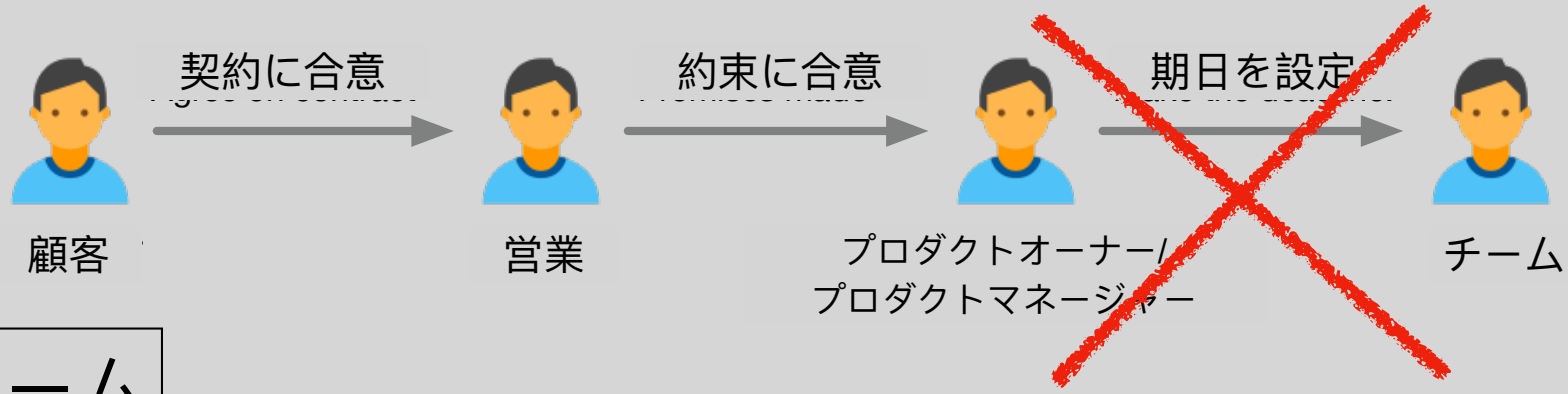
Prioritise different deadlines and reduce work in progress on high level

Item 1
Item 2
Item 3
Item 4
Item 5
...

Communicate priority and reasoning.

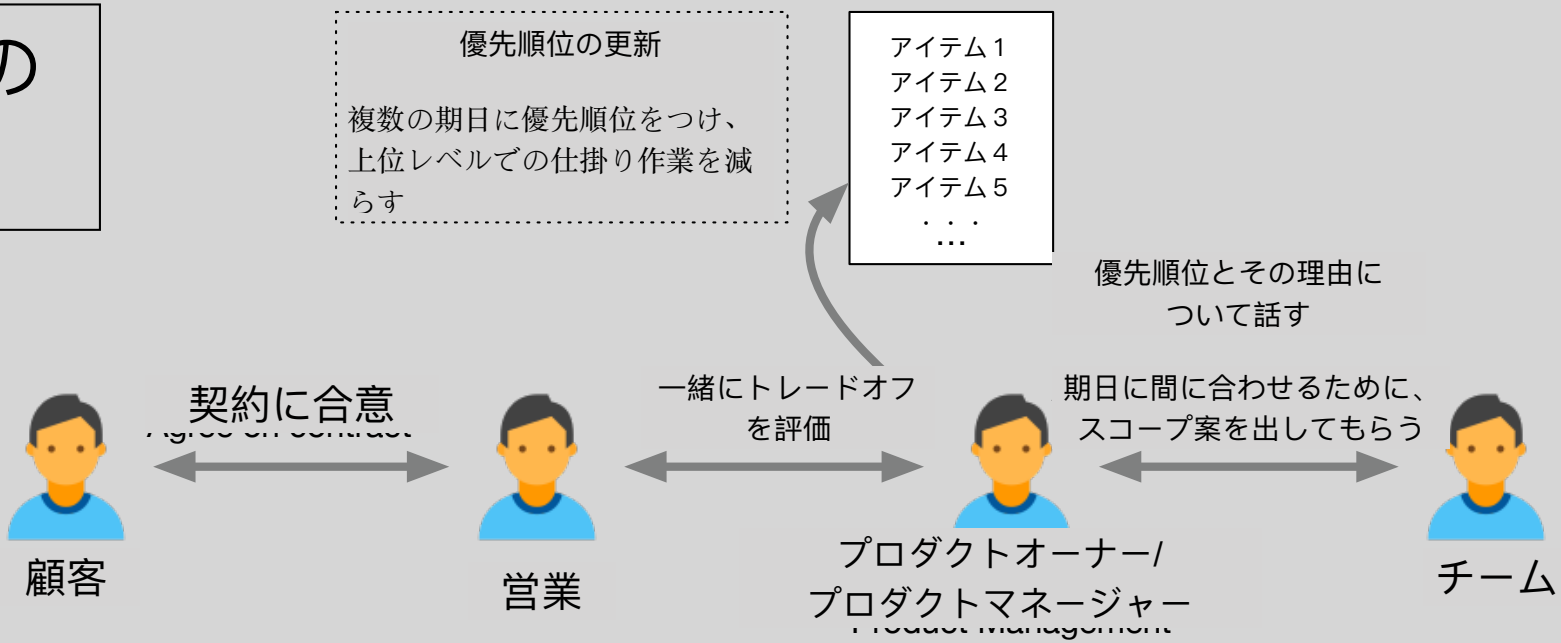


期日との付き合い方

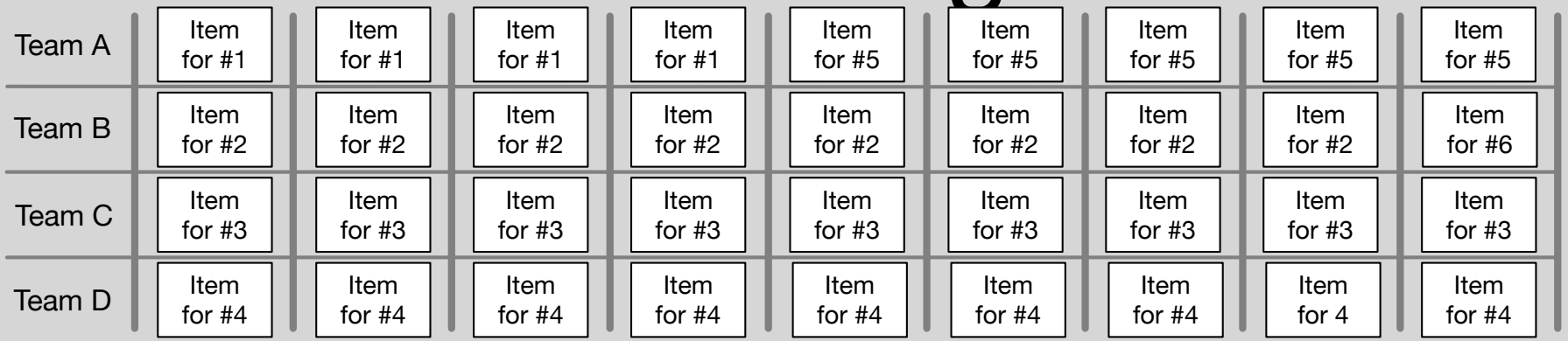


契約ゲーム

協働型の開発



Scheduling

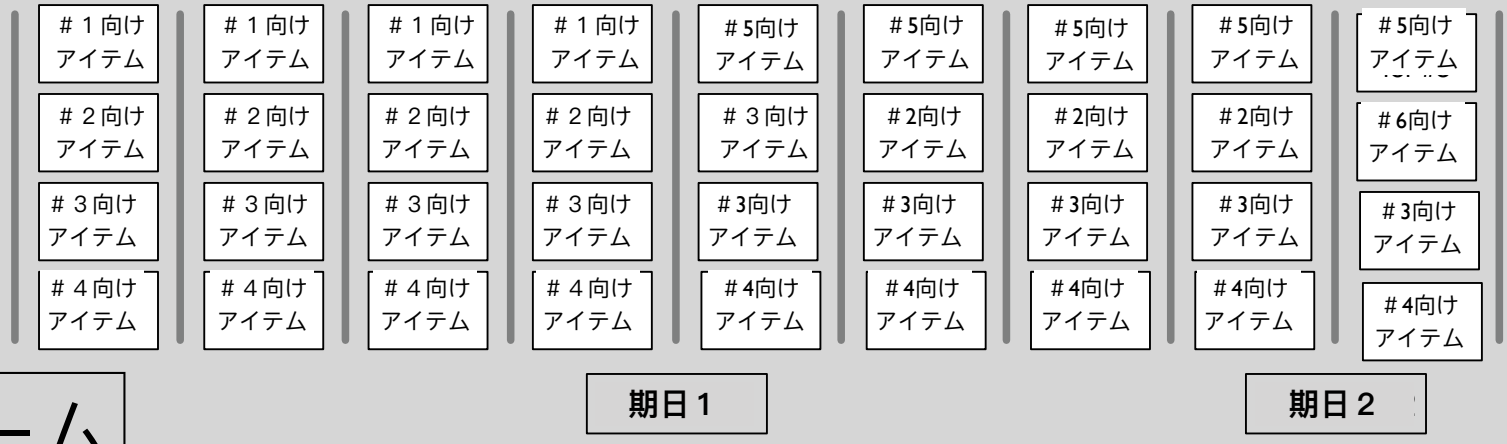


Contract game

Deadline 1

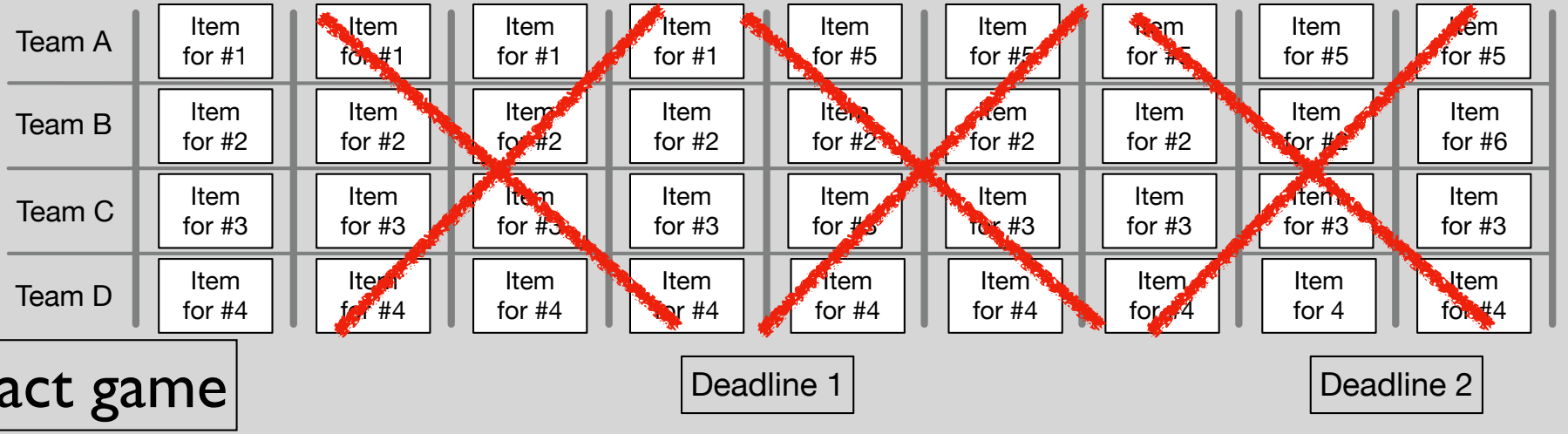
Deadline 2

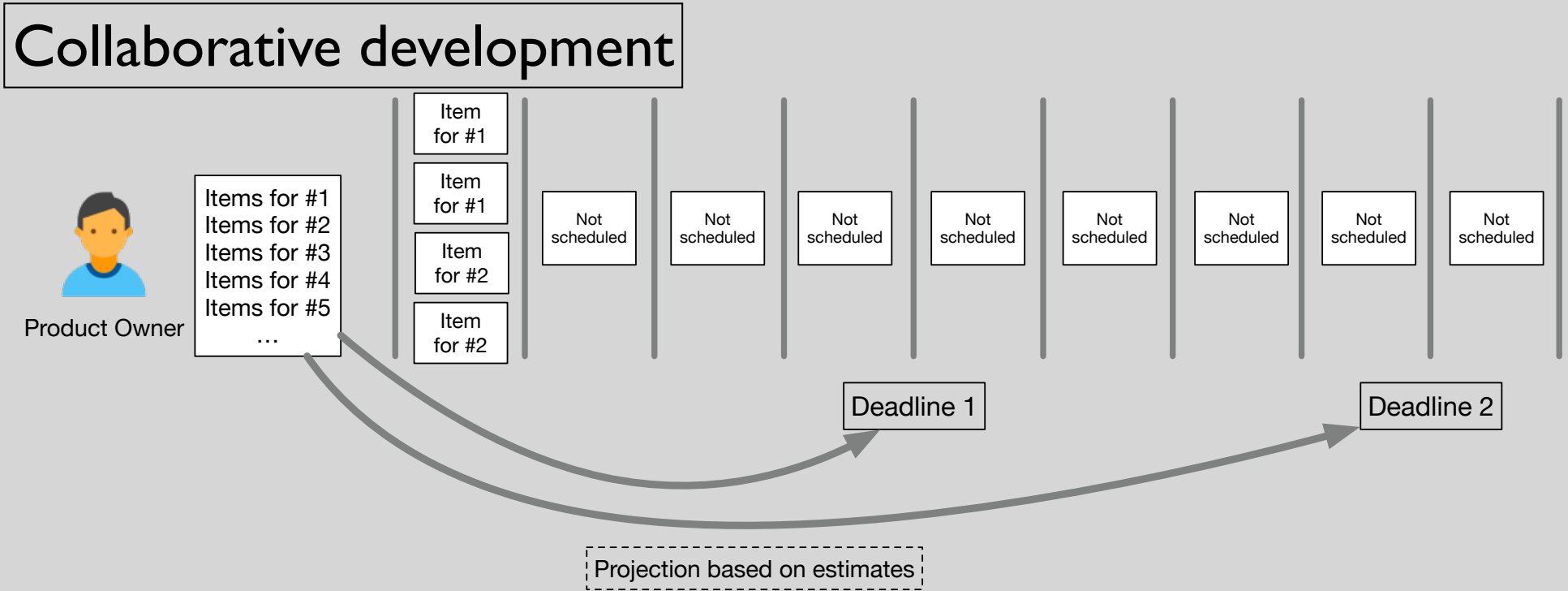
スケージュール



契約ゲーム

Scheduling


Team A
Team B
Team C
Team D
Item for #1
Item for #2
Item for #3
Item for #4
Item for #5
Item for #6
Item for 4
Contract game
Deadline 1
Deadline 2

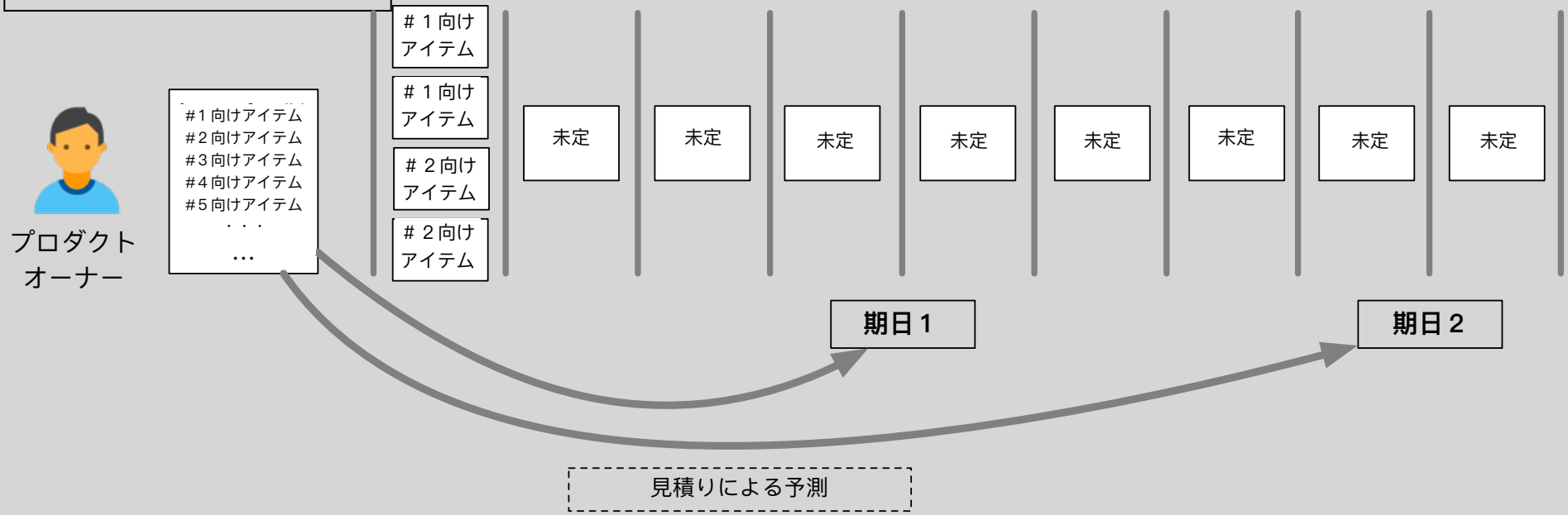

Collaborative development
Product Owner
Items for #1
Items for #2
Items for #3
Items for #4
Items for #5
…
Item for #1
Item for #1
Item for #2
Item for #2
Not scheduled
Deadline 1
Deadline 2
Projection based on estimates

スケージュール



契約ゲーム

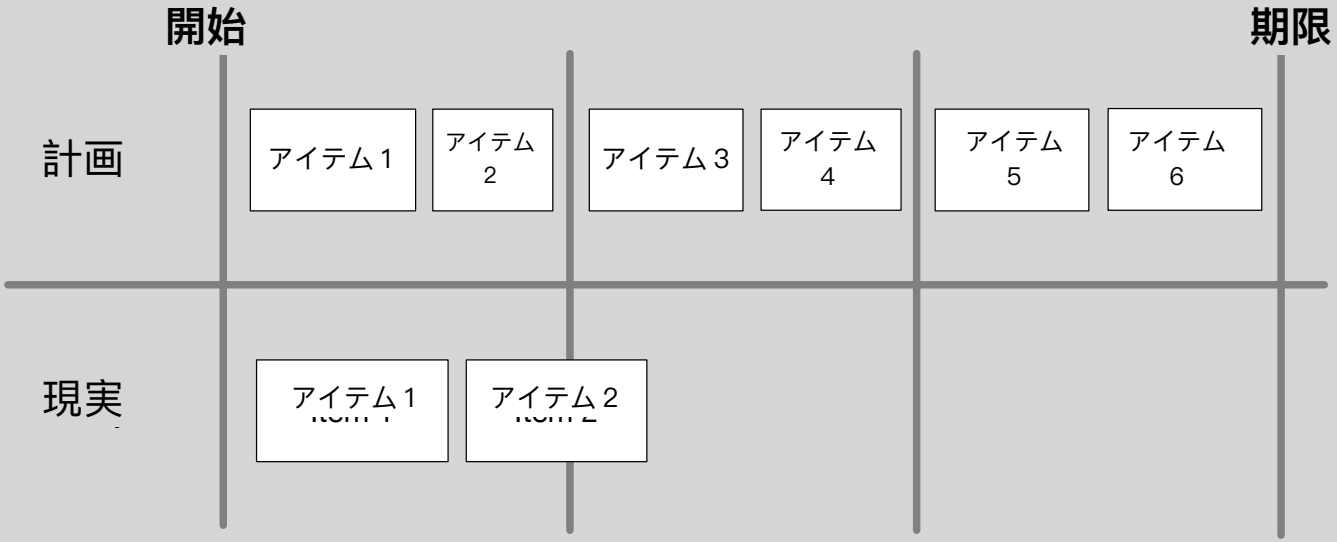
協働型の開発



契約ゲーム - スピルオー

なんで気にすの?
スピルオーバーなんて
気にしなくて良いんじゃん!

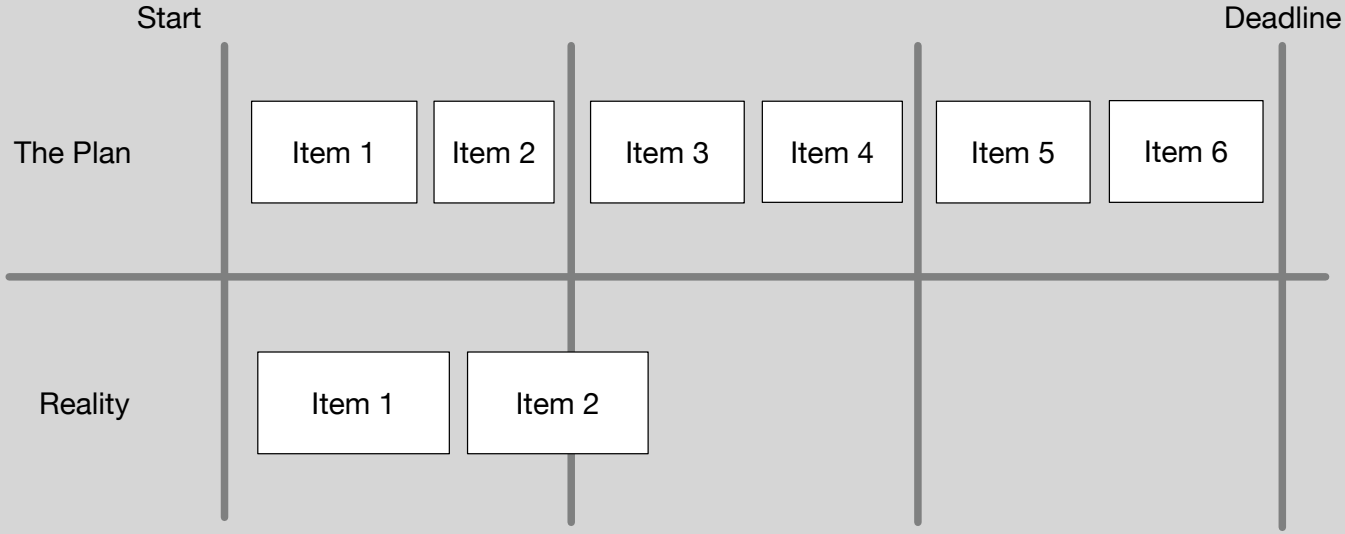
これらはイテレーションではない



Contract Game - Spillover

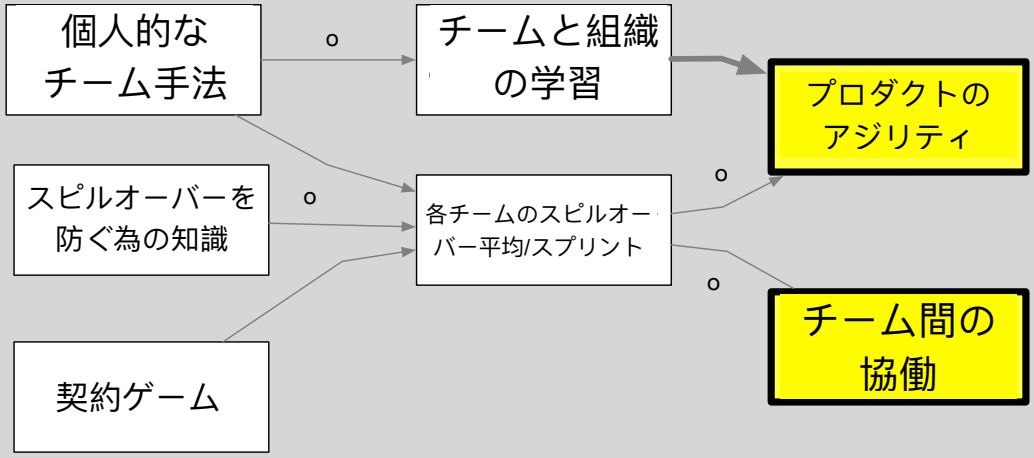
Why bother!
Spill-over, who cares!

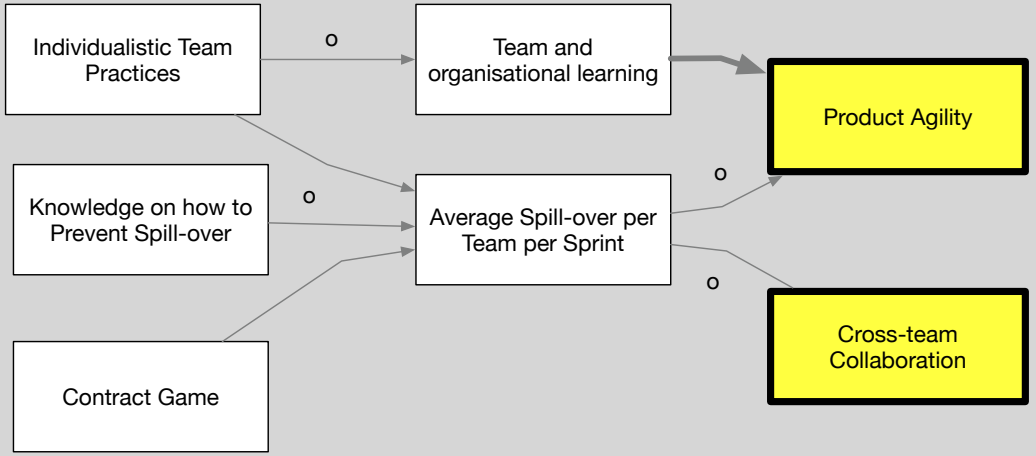
These iterations are fake



スピルオーバーの結果

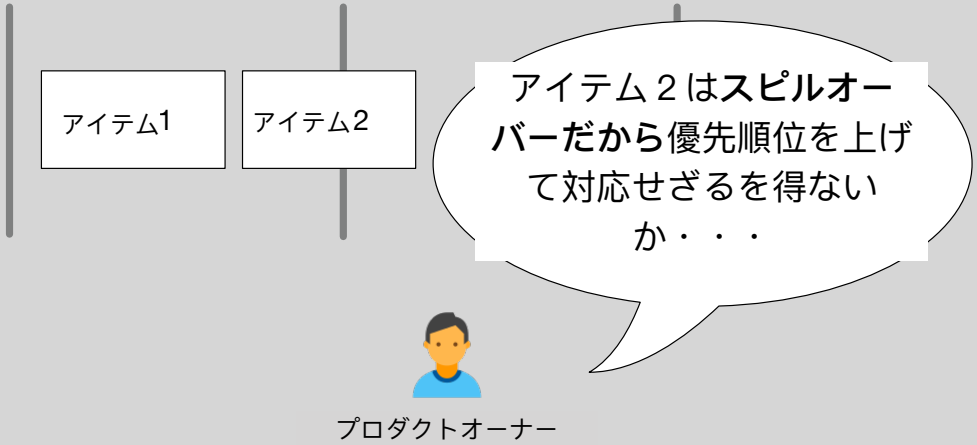
What causes spill-over





プロダクトアジリティ

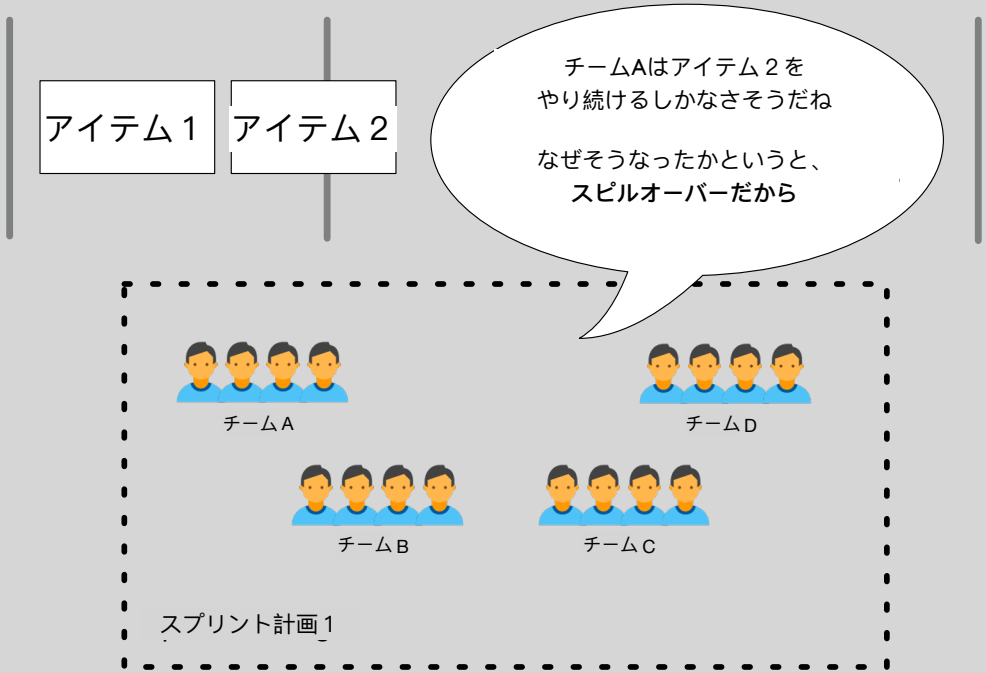
Product Agility

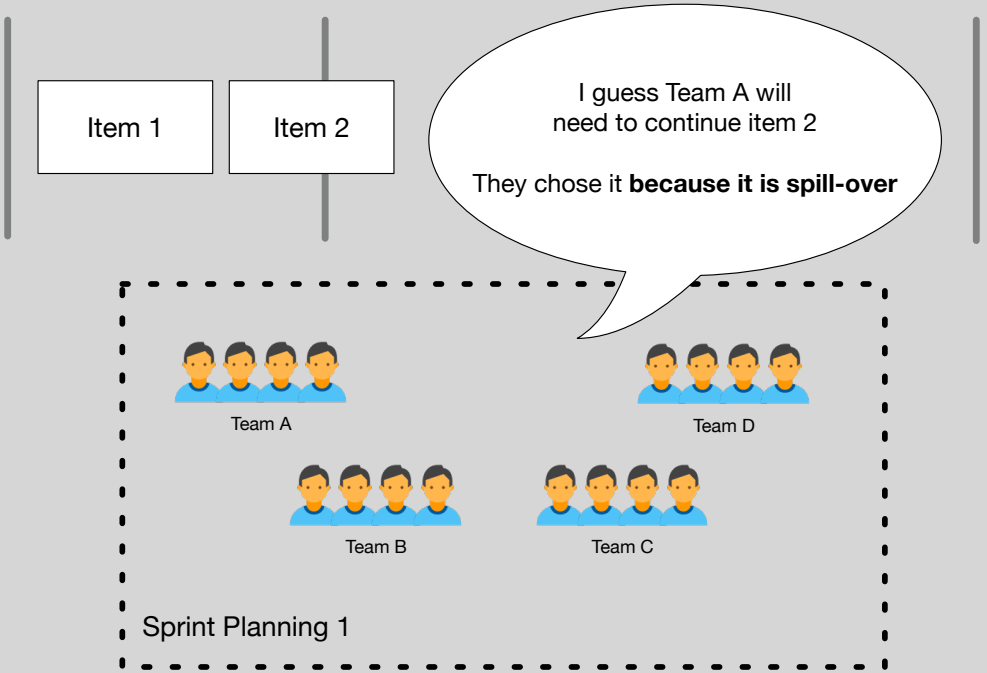




チーム横断コラボレーション

Cross-team Collaboration







まとめ（１）

- スピルオーバーの計測は単にスピルオーバーの計測だけでなく:
  - どの程度チームが協働できているか
  - 契約ゲームがおこっているか
  - チームがスプリント中の自分たちの選択と責任を理解しているか

Summary (1)

- Measuring spillover does not only measure spillover but also:
  - How well teams are collaborating
  - Whether there are contract game dynamics ongoing
  - Whether teams understand their choices and responsibilities during the Sprint



まとめ! (2)

- スピルオーバーを減らすようにすることは、下記のようなことにつながる:
  - チーム内の協働の向上
  - POの柔軟性の向上
  - チーム間の協働の向上

Summary! (2)

- Therefore, managing to lower spillover usually leads to:
  - Increased team collaboration
  - Increased PO flexibility
  - Increased cross-team collaboration



なぜそんなに
スピルオーバーに執着しているの?
Why are you always so obsessed with spill-over?

測定が容易な指標であり、これに注目することで、**チーム内およびチーム間の協働が向上**し、**健全なプロダクトオーナーシップ**につながる
It is an **easy to measure** metric and focus on it leads to **increased in- and cross-team collaboration** and **healthy Product Ownership**





12-14 October

https://less.works/course-details/certified-less-practitioner-tokyo-4457