



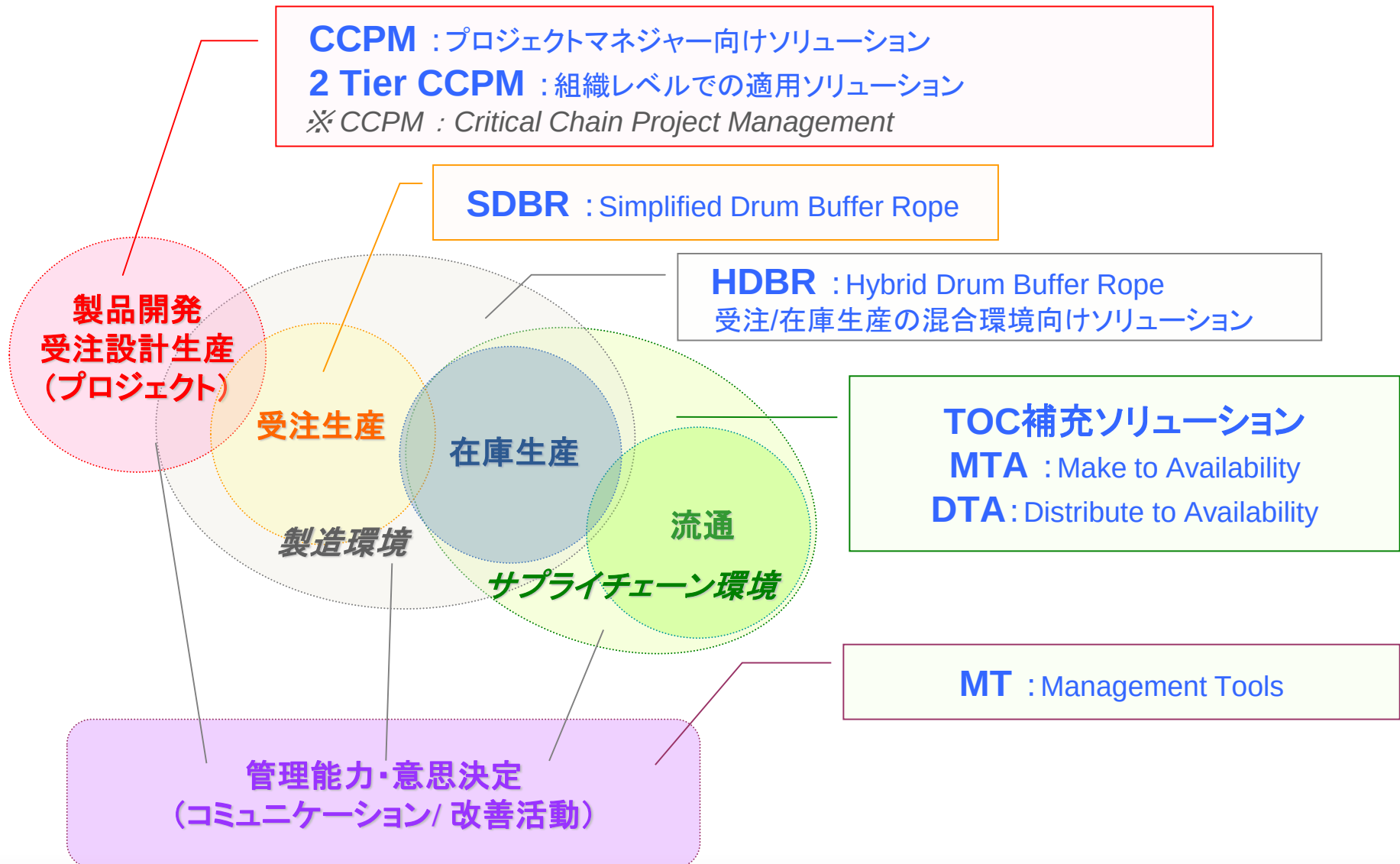
4th JTA/TOCPA JOINT CONFERENCE 2019

44th TOCPA International Conference

18 November 2019, Tokyo, Japan

プロジェクト環境（組織）における **TOC Solution — “2 Tier CCPM”**

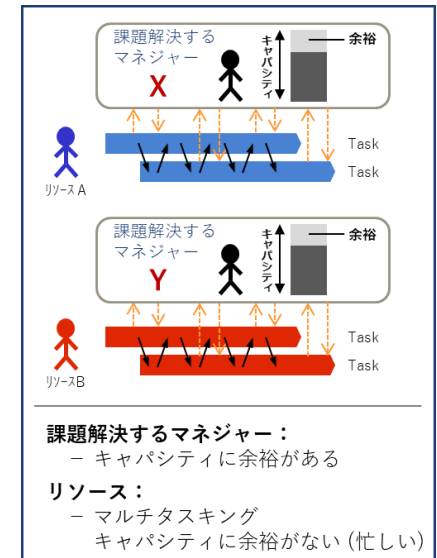




Original CCPM と 2 Tier CCPM の違い

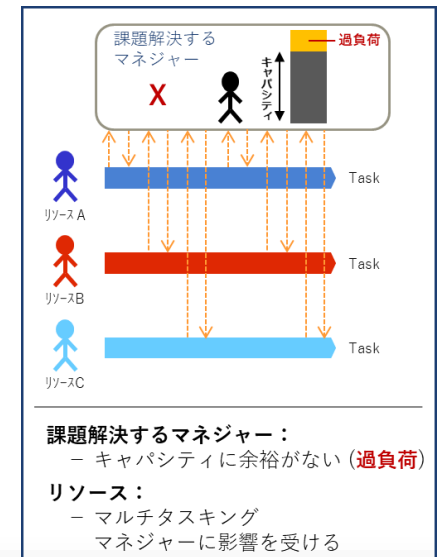
Original CCPM が機能する環境

- あるプロジェクト(1つずつ)のパフォーマンスを改善するためのソリューション
 - 根本問題は、“リソースのマルチタスキング”
- リソースや課題解決するマネジャーは、ある一定期間以上、あるプロジェクトに専属的に従事する。
 - 課題解決するマネジャー: キャパシティに余裕がある
 - リソース: キャパシティに余裕がない (忙しい)



2 Tier CCPM が機能する環境

- プロジェクト組織のパフォーマンスを改善するためのソリューション
 - 根本問題は、“課題解決するマネジャーのマルチタスキング”
- リソースや課題解決するマネジャーは、他のプロジェクトや非プロジェクト業務（問い合わせ対応、営業支援、アフターフォロー）などを同時に行わなければならない。
 - 課題解決するマネジャー: キャパシティに余裕がない (常に忙しい)
 - リソース: キャパシティに余裕がない/ある (マネジャーに影響を受ける)

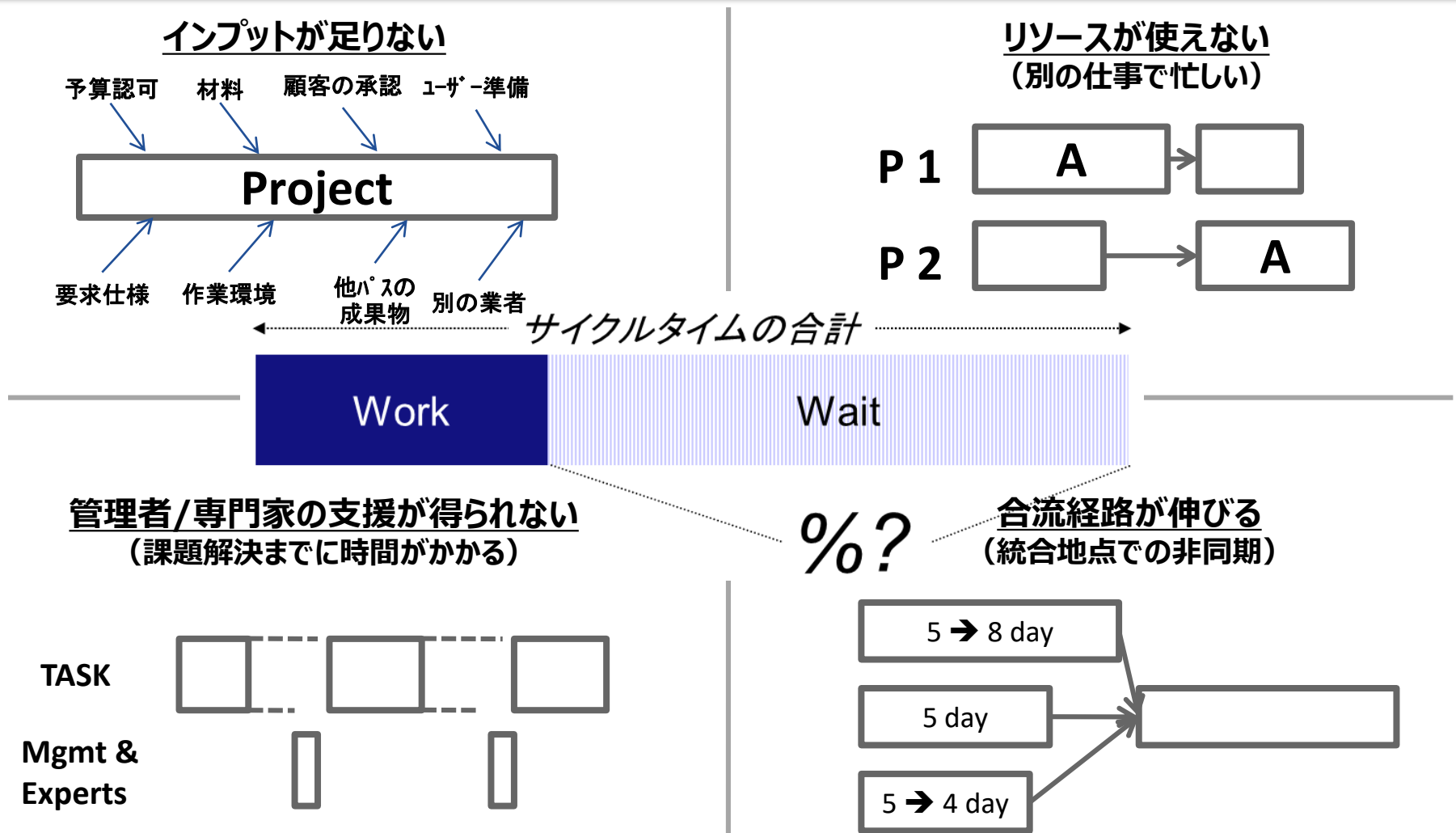


プロジェクト組織における典型的な問題症状

1. 必要な時に必要なリソースが使えないことが多い
2. プロジェクトの多くは予定より長くかかる
3. プロジェクトのスコープ/仕様がカットされることが多い
4. 進行中プロジェクトが「突発業務」によって遮断される
5. 中間締切りに間に合わせるための苦労が多い
6. 優先順位がころころ変わる
7. ぎりぎりで仕様が変更になって修正に追われる
8. やり直しが多すぎる
9. 残業が多すぎる
10. 予算に対してコスト超過することが多い



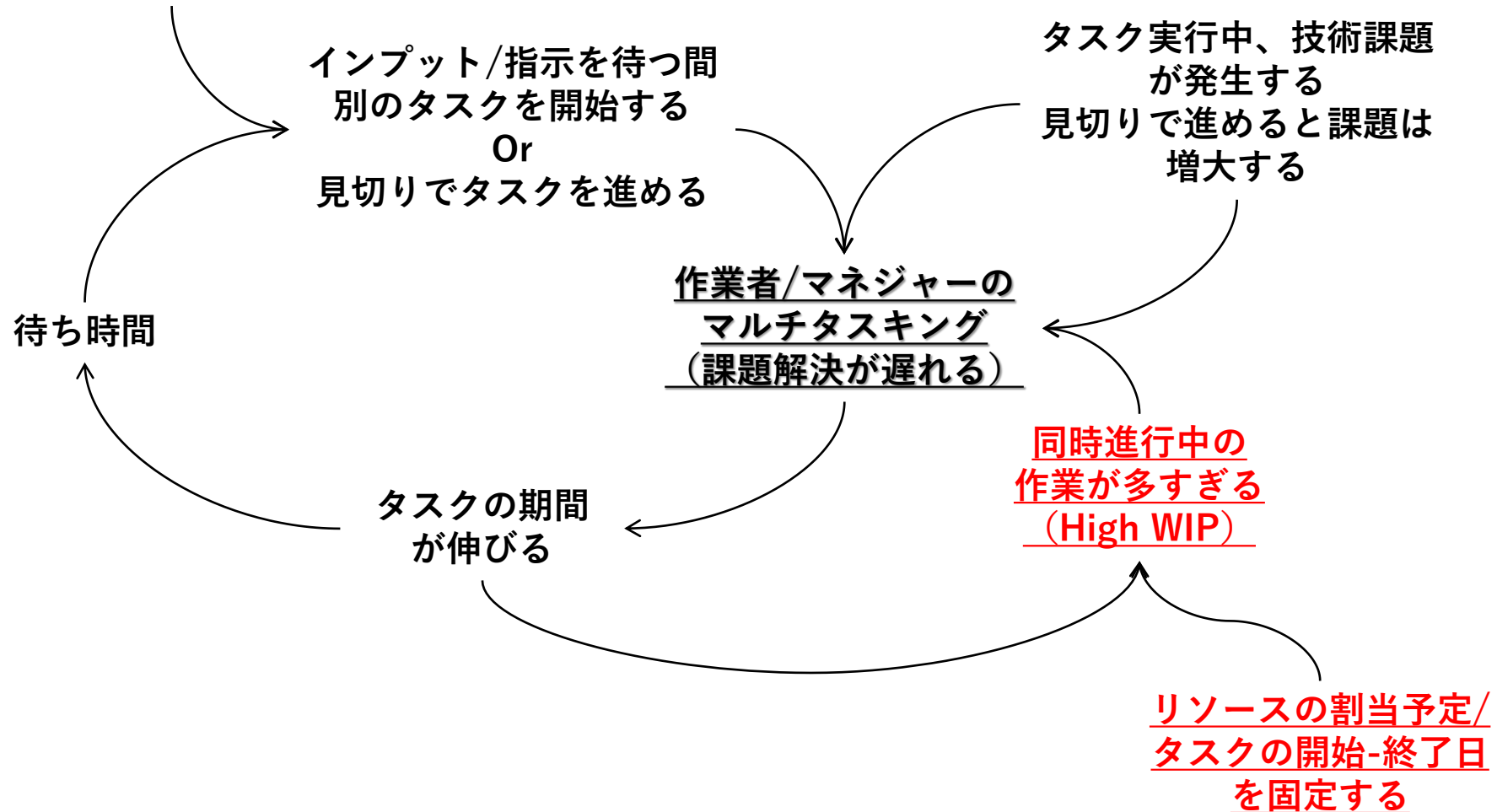
プロジェクトにおける“待ち時間”は？ – 主な4つの要因



**待ち時間が発生した際、
我われがとる典型的な行動は何だろうか？**

悪循環のサイクル

できるところから着手すれば
進捗するという思い込み
(手空き/空スペースはもったいない)



悪循環のサイクル (別の視点)

できるところから着手すれば
進捗するという思い込み

リソースの割当予定/
タスクの開始-終了日
を固定する

リソースは課題解決を待つ間
別のタスクを開始する、
見切りで進める
(High WIP)

課題解決までの
期間が長くなる
(課題解決が遅れる)

マネジャーの
マルチタスキング

マネジャーは
対処すべき課題を
多く抱える
(High WIP)

突発的な
仕様確認、問合せ
対応が発生する

課題の発生と同時に
マネジャーは即座に
課題解決に対応する

タスク進行中
課題は必ず発生する
(業務間の整合、リソースへの支援)
見切りで進めると課題は増大する

課題解決するマネジャーのマルチタスキング

Management and Support Resources マネジメントとサポートのリソース



- 意思決定や課題解決待ちのために、タスクが止まる
- 何をやるにしても時間がより多くかかる
- こういった中断は、リソースがないよりも よほど 性質が悪い

問題解決の方向性

何を変えるのか？

- High WIP
 - 薄く伸ばしたリソース割り当て
 - 進行中のプロジェクト/タスク数が多すぎる
- 固定的なリソース配置
 - リソースに仕事を割り当てる
- できるところから進捗を見せなければならない
 - タスクを開始できる条件が揃えば作業を開始する
 - タスク毎の開始-終了日に基づく局所的な優先順位



何に変わるのか？

- WIPの制限 (2-Tiered Planning)
 - タスクを2階層で構成し、Tier1タスクにリソースを集中させる
 - キャパシティに従い、進行中のTier1タスク数を制限する
- フレキシブルなリソース配置
 - 仕事にリソースを割り当てる
- 優先順位と利用可能キャパシティに従い、動的に投入
 - タスクを完了できる条件が揃わなければ作業を開始しない
 - バッファの消費状態に基づく単一の優先順位
 - 利用可能キャパシティに従い、動的にタスクを投入する

2 Tier CCPM の3つのルール

RULE 1: **2-Tiered Planning**

リソースを集中させ、進行中Tier1タスクを減らす

RULE 2: **IP Task Management**

Tier1タスク短縮のため
Tier2タスクにリソースを柔軟に配置する

RULE 3: **Queue Task Control**

優先順位と利用可能キャパシティに従って
Tier1タスクを動的に投入する

Rule 1: 2-Tiered Planning

（タスクを2階層で構成し、
Tier1 タスクにリソースを集中させる）

現行の典型的なPlanning 例

タスク名	リソース名	期間	1	2	3	4	5	6	7	8	9	10	11	12
詳細設計														
機能A														
機能A-x														
機能A-x-1 設計	メンバB	2												
機能A-x-2 設計	メンバA	1												
機能A-x-3 設計	メンバB	1												
機能A-x-4 設計	メンバB	2												
機能A-x-5 設計	メンバA	1												
機能A-x-6 設計	メンバB	2												
機能A-x-7 設計	メンバB	1												
機能A-x-8 設計	メンバA	1												
機能A-x-9 設計書まとめ	メンバB,メンバA	1												
機能A-z														
機能A-z-1 設計	メンバD	1												
機能A-z-2 設計	メンバD	1												
機能A-z-3 設計	メンバD	1												
機能A-z-4 設計	メンバD	1												
機能A-z-5 設計	メンバD	1												
機能A-z-6 設計	メンバD	1												
機能A-z-7 設計	メンバD	1												
機能A-z-8 設計書まとめ	メンバD	2												

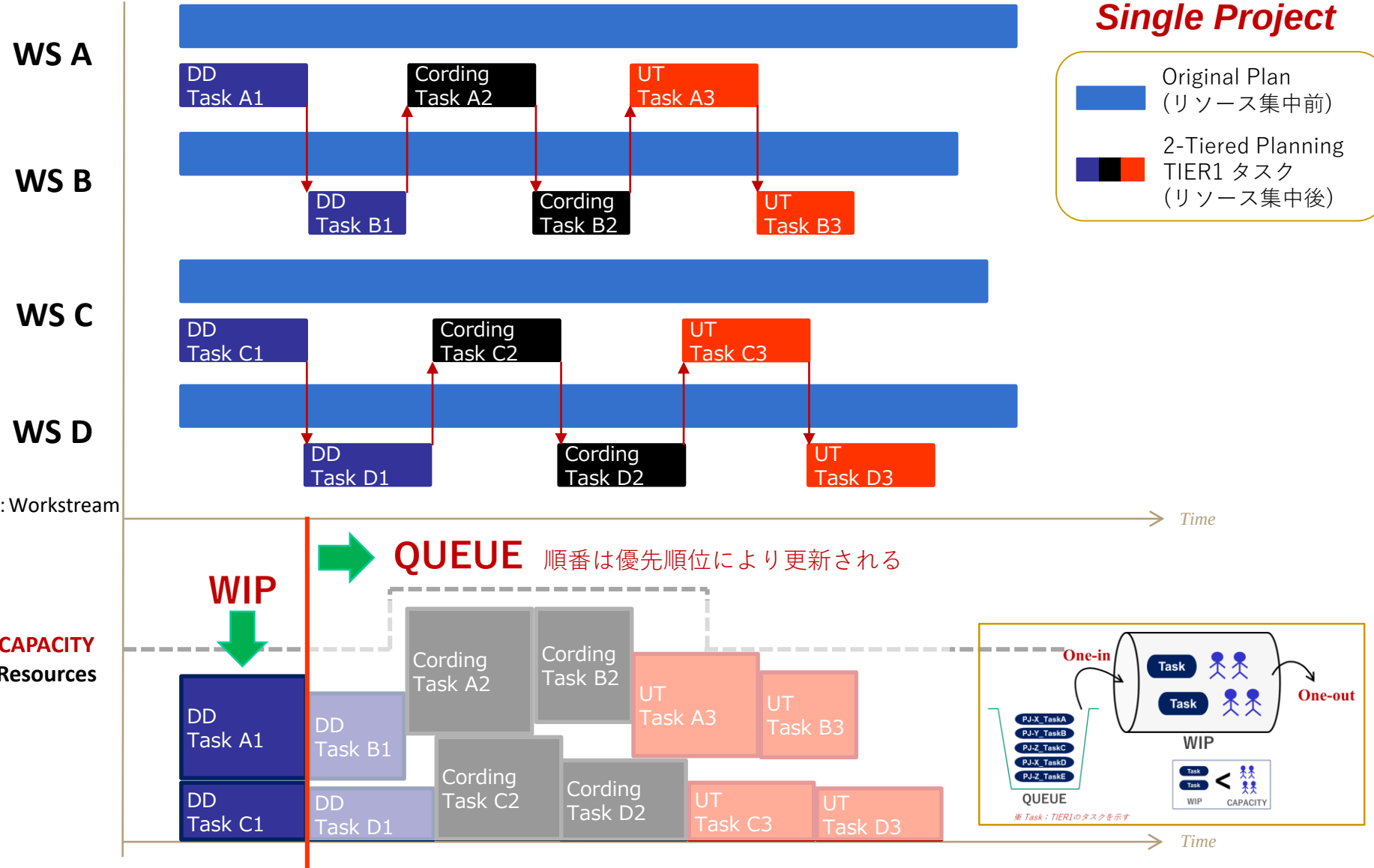
- *TIER1 Tasks WIP : 2*
- *Sub tasks(TIER2) WIP : 3*

2-Tiered Planning 例

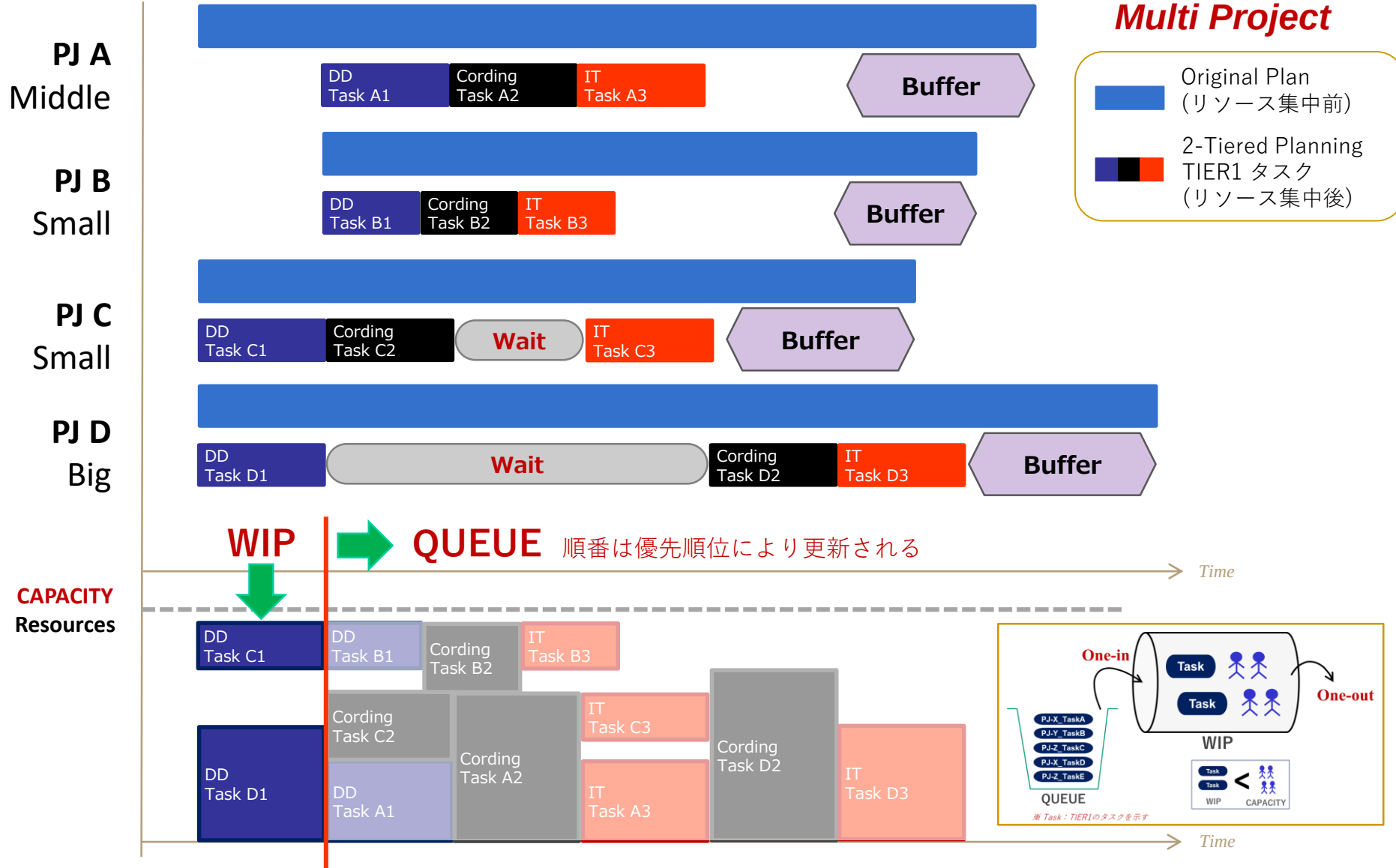
タスク名	リソース名	期間	1	2	3	4	5	6	7	8	9	10	11	12
詳細設計														
機能A														
機能A-x														
機能A-x-1 設計	メンバB	2												
機能A-x-6 設計	メンバB	2												
機能A-x-3 設計	メンバD	1												
機能A-x-4 設計	メンバD	2												
機能A-x-7 設計	メンバD	1												
機能A-x-2 設計	メンバA	1												
機能A-x-5 設計	メンバA	1												
機能A-x-8 設計	メンバA	1												
機能A-x-9 設計書まとめ	メンバB,A,D	1												
機能A-z														
機能A-z-1 設計	メンバD	1												
機能A-z-2 設計	メンバD	1												
機能A-z-3 設計	メンバD	1												
機能A-z-4 設計	メンバA	1												
機能A-z-5 設計	メンバA	1												
機能A-z-6 設計	メンバB	1												
機能A-z-7 設計	メンバB	1												
機能A-z-8 設計書まとめ	メンバB,A,D	2												

- *TIER1 Tasks WIP : 1*
- *Sub tasks(TIER2) WIP : 3*

Rule 1: リソースを集中させ、進行中Tier1タスクを減らす



Rule 1: リソースを集中させ、進行中Tier1タスクを減らす

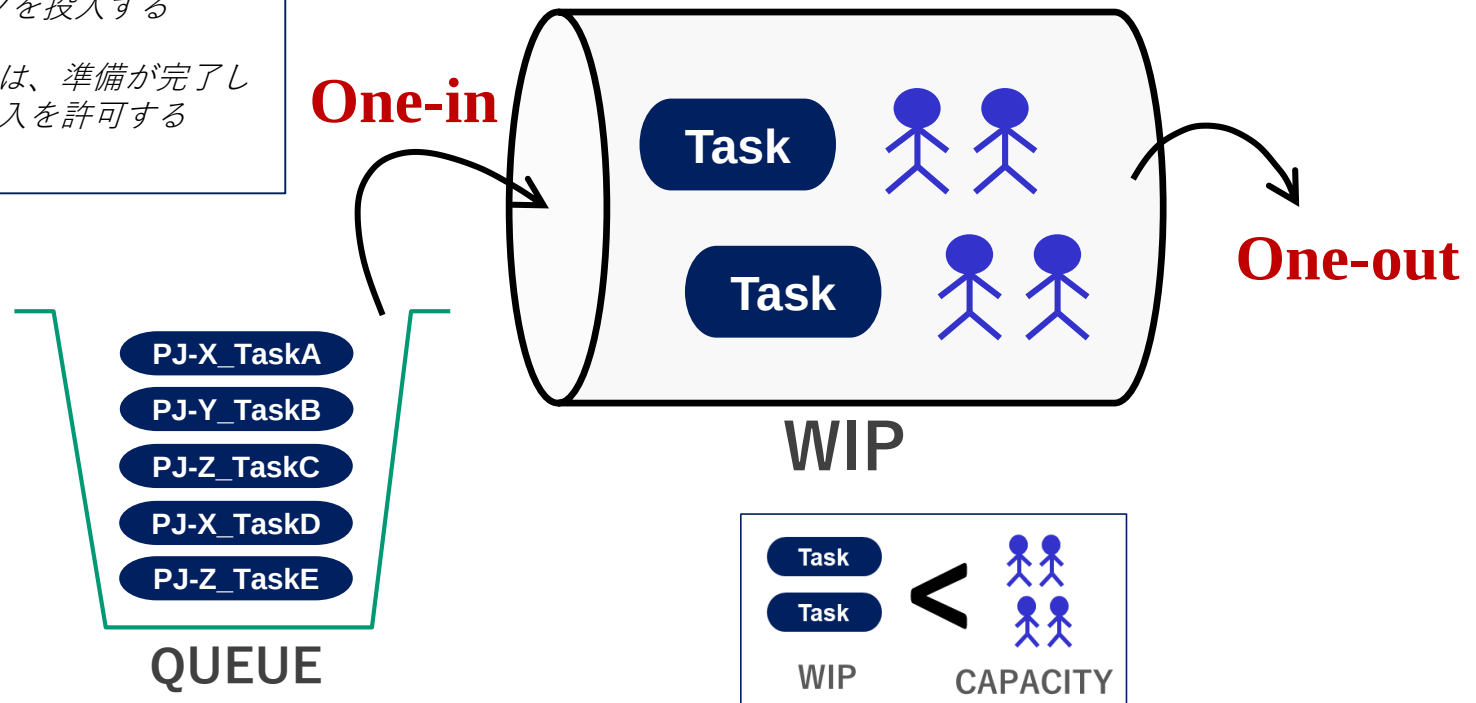


Rule 1: PIPELINING – FOCUS and FINISH

キャパシティを超えないよう、タスクの開始をずらす

- リソースのキャパシティ以上にWIPを増やさない
- WIP中のタスクが完了した後、QUEUEのタスクを投入する
- 投入するタスクは、準備が完了したタスクのみ投入を許可する(フルキット)

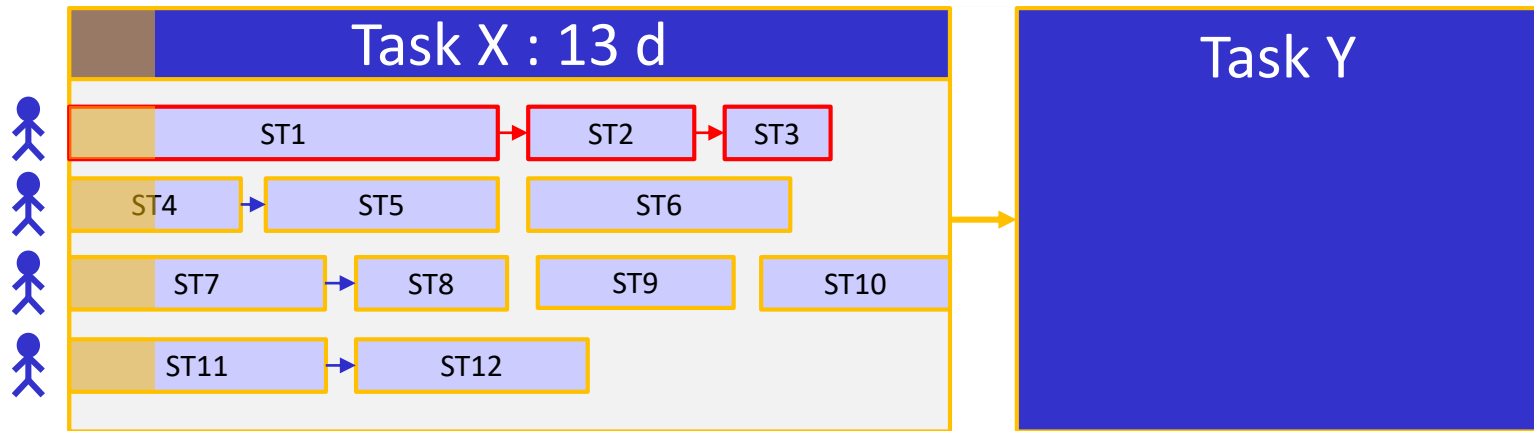
PIPELINING – FOCUS AND FINISH



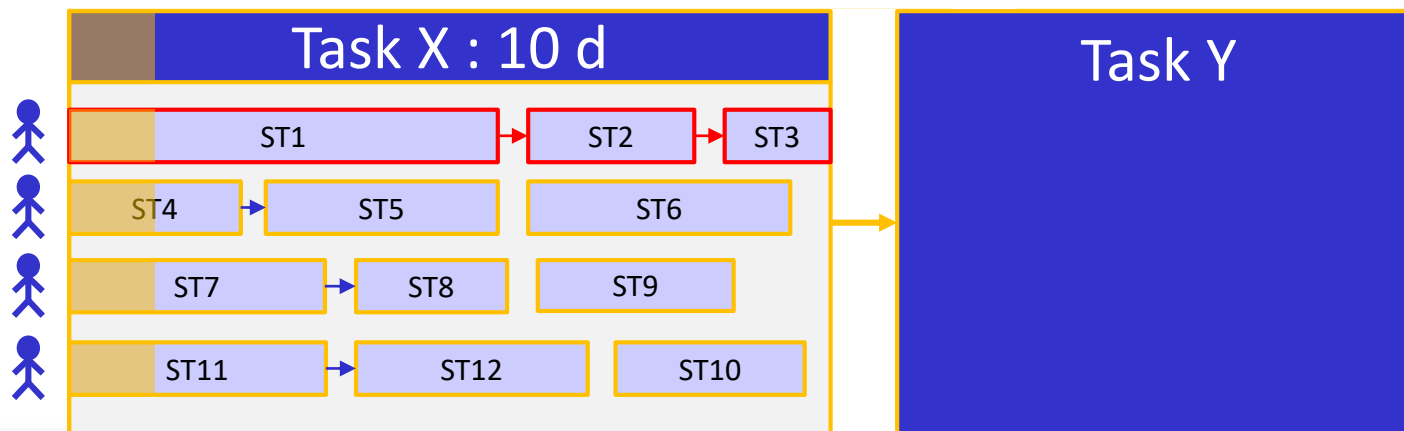
※ Task : TIER1のタスクを示す

Rule 2: Tier2タスクにリソースを柔軟に配置する

人に仕事を割り当てる (初めに決めたリソース割当を固定)



仕事に人を割り当てる (TIER1 タスクを短縮するようリソース配置を柔軟に行う)



Rule 2: IP Task Management

PIPELINING – FOCUS AND FINISH

- リソースのキャパシティ以上にWIPを増やさない
- WIP中のタスクが完了した後、QUEUEのタスクを投入する
- 投入するタスクは、準備が完了したタスクのみ投入を許可する (フルキット)

One-in

One-out

PJ-X_TaskA

PJ-Y_TaskB

PJ-Z_TaskC

PJ-X_TaskD

PJ-Z_TaskE

QUEUE

Task

Task

WIP

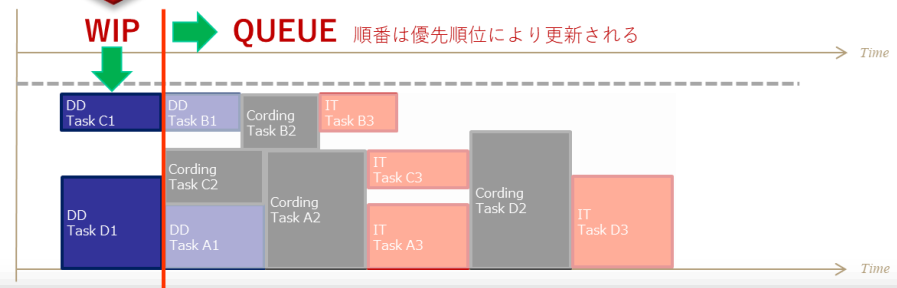
Rule 2

WIP

QUEUE

順番は優先順位により更新される

**CAPACITY
Resources**



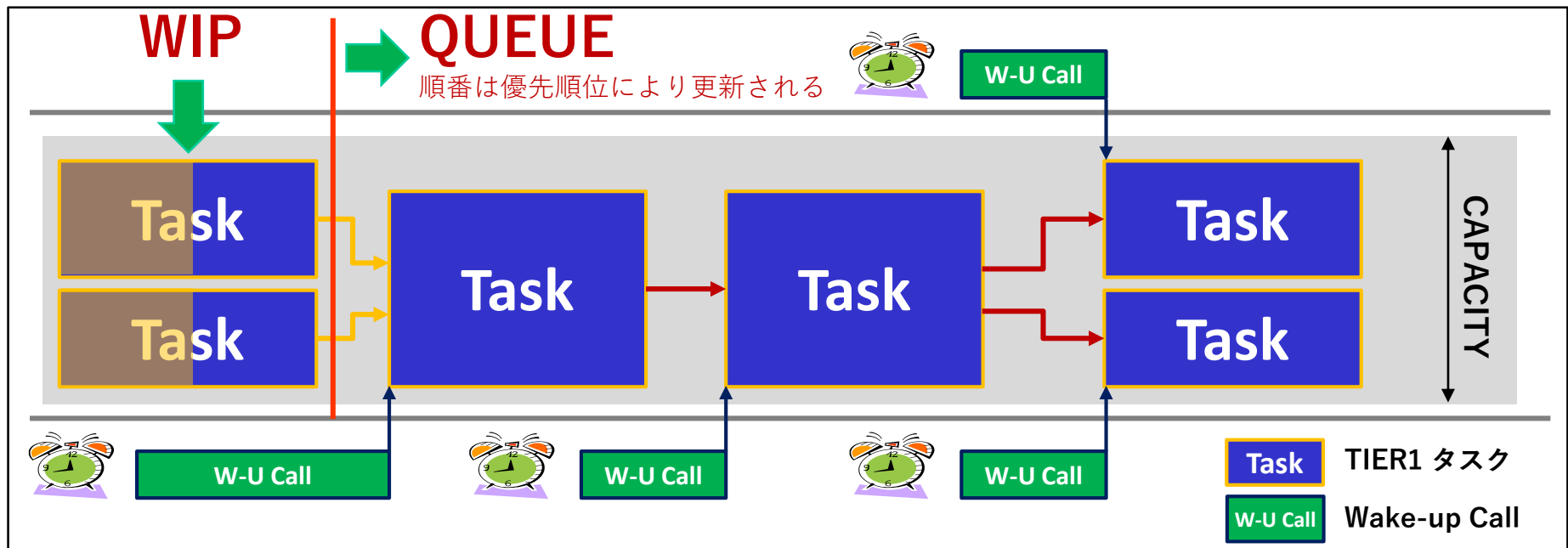
※ Task : TIER1のタスクを示す

Rule 3: TIER1 タスクのインプットとリソースの利用可能状態

QUEUE - TIER1 タスクのインプットとリソースの利用可能状態の確認

マネジャーは、IP - TIER1 タスクの進捗状況を確認し、QUEUE - TIER1 タスクの担当予定リソースの状態とその準備状況(フルキット)を確認させるようにする。

このことを、“**Wake-up Call**”と呼ぶ



Wake-up Call は、“リソース”と“その準備”が対象である

※準備とは、**TIER1 タスクを完了できるインプット** が揃っているかどうかである

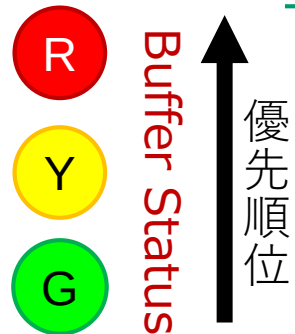
Rule 3: Queue Task Control

PIPELINING – FOCUS AND FINISH

- リソースのキャパシティ以上にWIPを増やさない
- WIP中のタスクが完了した後、QUEUEのタスクを投入する
- 投入するタスクは、準備が完了したタスクのみ投入を許可する(フルキット)

One-in

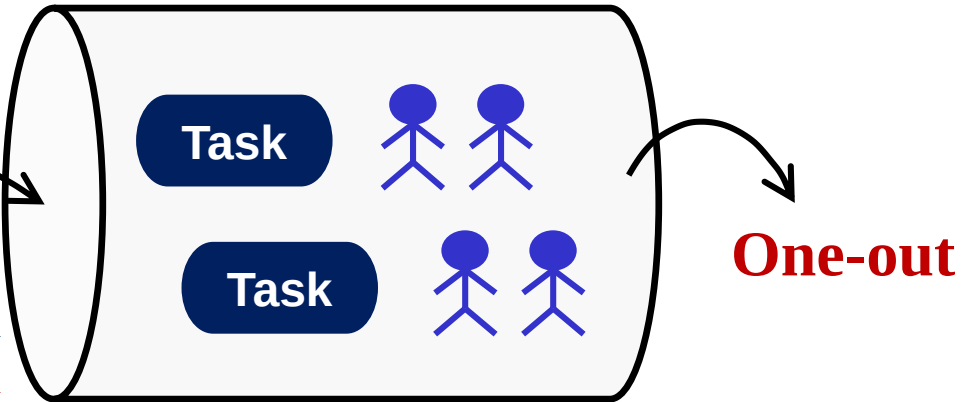
One-out



PJ-X_TaskA Input - OK
PJ-Y_TaskC Input - OK
PJ-Z_TaskD Input - OK
PJ-X_TaskB Input - NG
PJ-Z_TaskE Input - NG

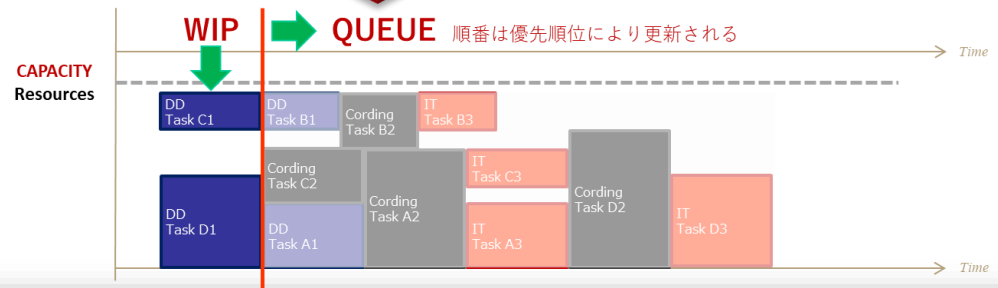
QUEUE

Rule 3



WIP

Rule 3



※ Task : TIER1のタスクを示す

「2 Tier CCPM」が組織にもたらすもの

生産性と スピード

- スループットの向上
- リードタイムの短縮
- コスト削減

信頼性

- 高い納期順守率
- 仕様を満たす能力
- コストの予実差異を小さくする

管理能力

- 素早い課題解決(火消し対応が少なくなる)
- 明確な優先順位により、管理をやり易くする
(マネジャーの過負荷の解消)