

リーン開発について

～CCPMとアジャイル開発でプロジェクトを成功に導く～

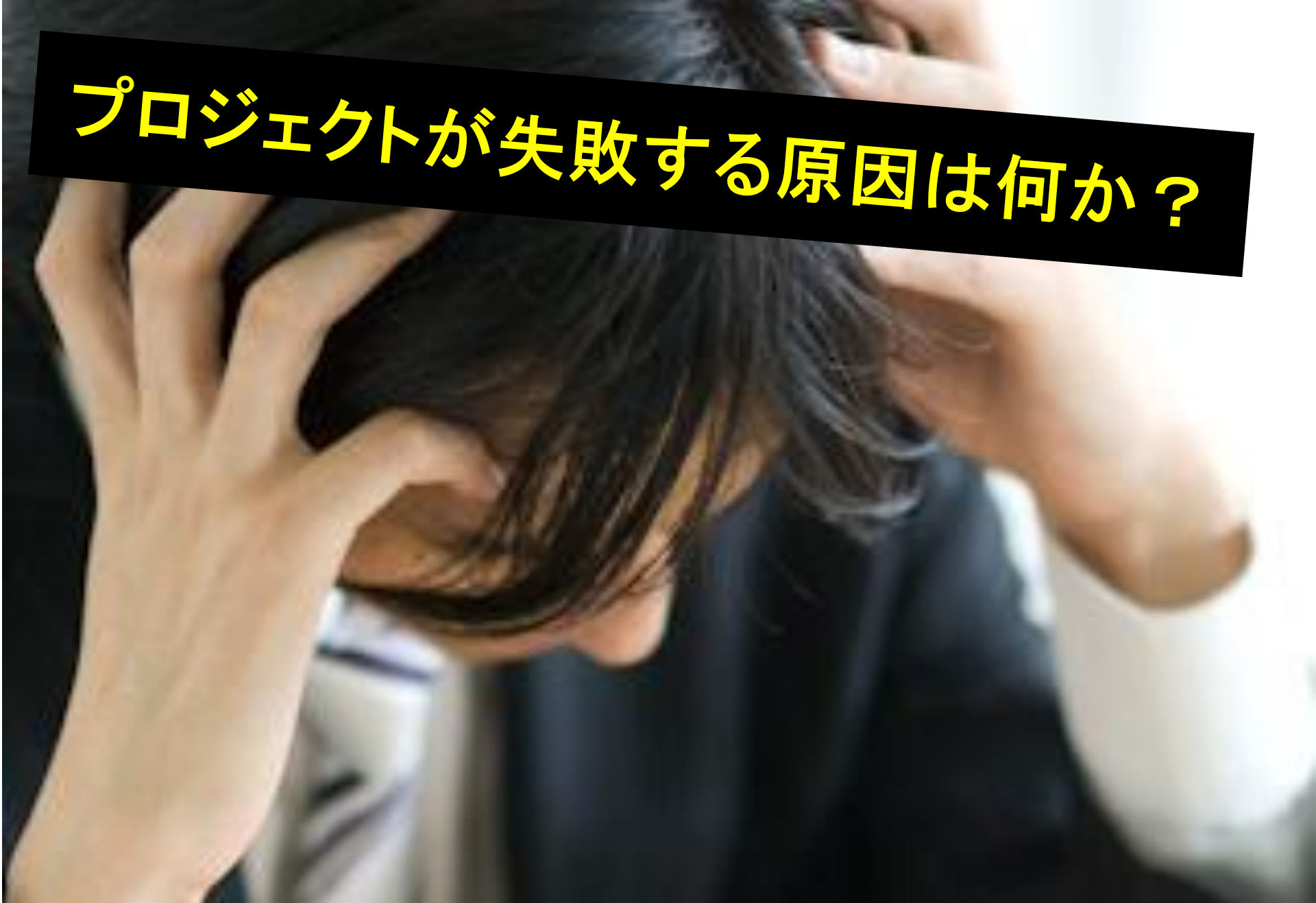
小野里 啓志 (Satoshi Onozato)

ITプロジェクトの 成功率 ???. ?%

日経コンピュータ2003.11.17
1,746社回答(12,456社中:有効回答率13.9%)
QCD(品質・コスト・納期)をクリアできたプロジェクト

ITプロジェクトの 成功率 26.7%

日経コンピュータ2003.11.17
「特集プロジェクト成功率は26.7%」より
1,746社回答(12,456社中:有効回答率13.9%)



プロジェクトが失敗する原因は何か？

プロジェクトマネジメントの特徴

- 基本的には人間相手の仕事
- 同じ仕事は二度とない
- 多くの人による分業作業
- 人の資質に大きく依存
- 多くの場合、きわめて高い不確実性がある



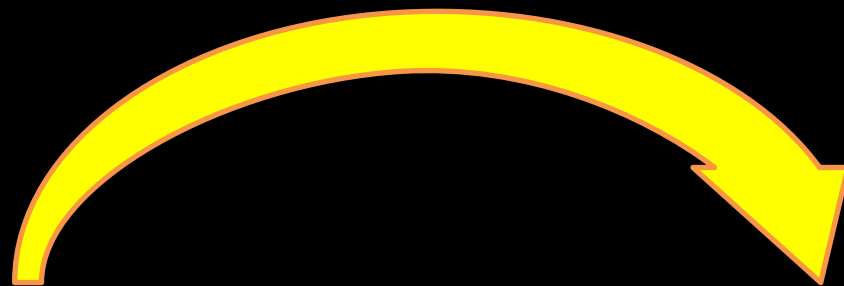
A close-up photograph of a man with dark hair, wearing a dark suit, white shirt, and a striped tie. He is looking down and to the right with a thoughtful expression, his hand resting under his chin.

プロジェクトが失敗する原因は何か？

不確実性
"ゼロ"にはできない

今日お話しすること

**不確実性がもたらす
ムダを解決**



スケジュール遅延

プロジェクト成功

POINT

～不確実性がもたらすムダを解決～

1. リーンソフトウェア開発

- ムダを省いたソフトウェア開発手法
- トヨタ生産方式からムダ取りの視点を学ぶ

2. CCPMスケジューリング

- 不確実性を想定したバッファマネジメント

本日の流れ

リーン
ソフトウェア開発

リーン開発に学ぶ
CCPMの改善点



CCPM
スケジューリング

QA

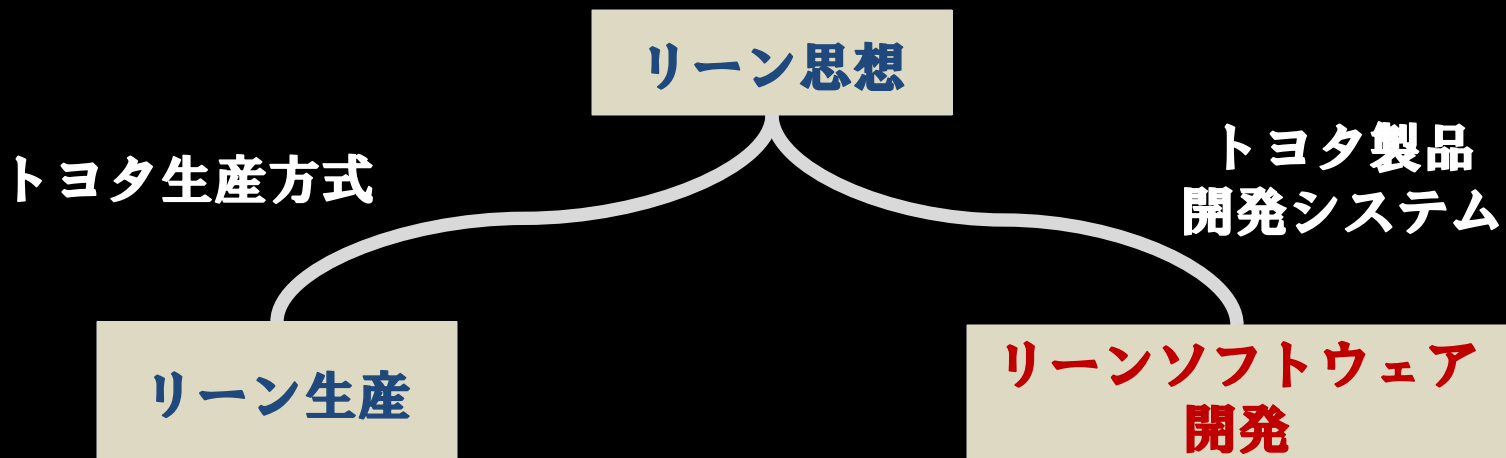
リーンソフトウェア 開発

リーンソフトウェア開発



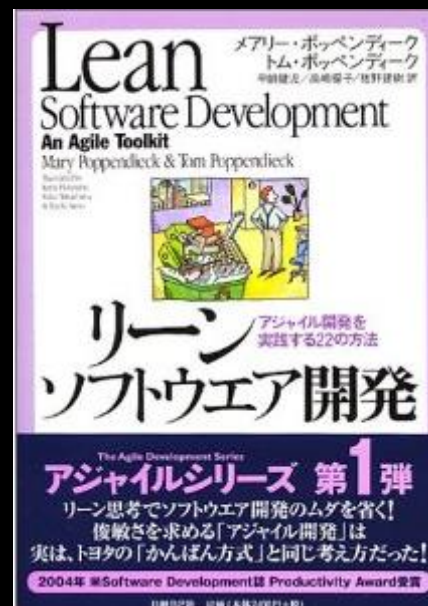
- ・ アジャイル方法論の1つ
- ・ 2003年、メアリー/トム ポップエンディークが共著
- ・ 製造業の「リーン生産方式」(トヨタ生産方式を含む)と、シックスシグマ品質イニシアティブをミックスした原則集
- ・ アジャイル開発が何故うまくいくのか、生産革新で実証済みの考え方の延長とマッピング

～思想の原点～



リーン思考の7原則

1. ムダをなくす
2. 品質を作り込む
3. 知識を作り出す
4. 決定を遅らせる
5. 速く提供する
6. 人を尊重する
7. 全体を最適化する



シンプルなものを複雑にするのはよくあること。
複雑なものを素晴らしくシンプルにすること、
それがクリエイティビティだ。

チャールズ・ミンガス



■リーン開発の根本

- ・ムダの排除

■リーン開発の実践

1. ムダを認識する
2. ムダの出所を見つけ出す

ムダを認識する

トヨタ生産方式

在庫のムダ

作りすぎのムダ

加工そのもののムダ

運搬のムダ

動作のムダ

手待ちのムダ

不良を作るムダ

ソフトウェア開発

未完成の作業のムダ

余分な機能のムダ

再学習のムダ

引き継ぎのムダ

タスク切り替えのムダ

遅れのムダ

欠陥(バグ)を作るムダ

やってみよう！
～不確実性がもたらすムダとは～

～年賀状送付～

期間 : 12/1 ～25

枚数 : 100枚

作業 : 1) 印刷

: 2) 一言手書きで記入

※10枚／日が限度

: 3) 投函

いつ、何枚、年賀状を作成し 投函しますか？

期間 : 12/1 ~25

枚数 : 100枚

作業 : 1) 印刷
: 2) 一言手書きで記入
 ※10枚／日が限度
: 3) 投函

日付	作業枚数		
	印刷	記入	投函
12/1	100		
12/2~11		10	
...			
12/24			100
12/25			

あなたは、どのパターンですか？

開始

12/1

10

20

×切

25

A	100		
	10 × 10		100
B		100	
		10 × 10	
C		10 × 10	
		10 × 10	
D	10 × 10		
	10 × 10		
	10 × 10		

予定通り
作業が進めば
うまくいく





予定通りには
進まない

プリンターのインク切れ発生！

	12/1	10	20	25
A	100 10 ... 10			100
B	最大100枚やり直し	100 10 ... 10 10 ... 10		
C	最大10枚やり直し	10 ... 10 10 ... 10 10 ... 10		
D	10 ... 10 10 ... 10 10 ... 10			

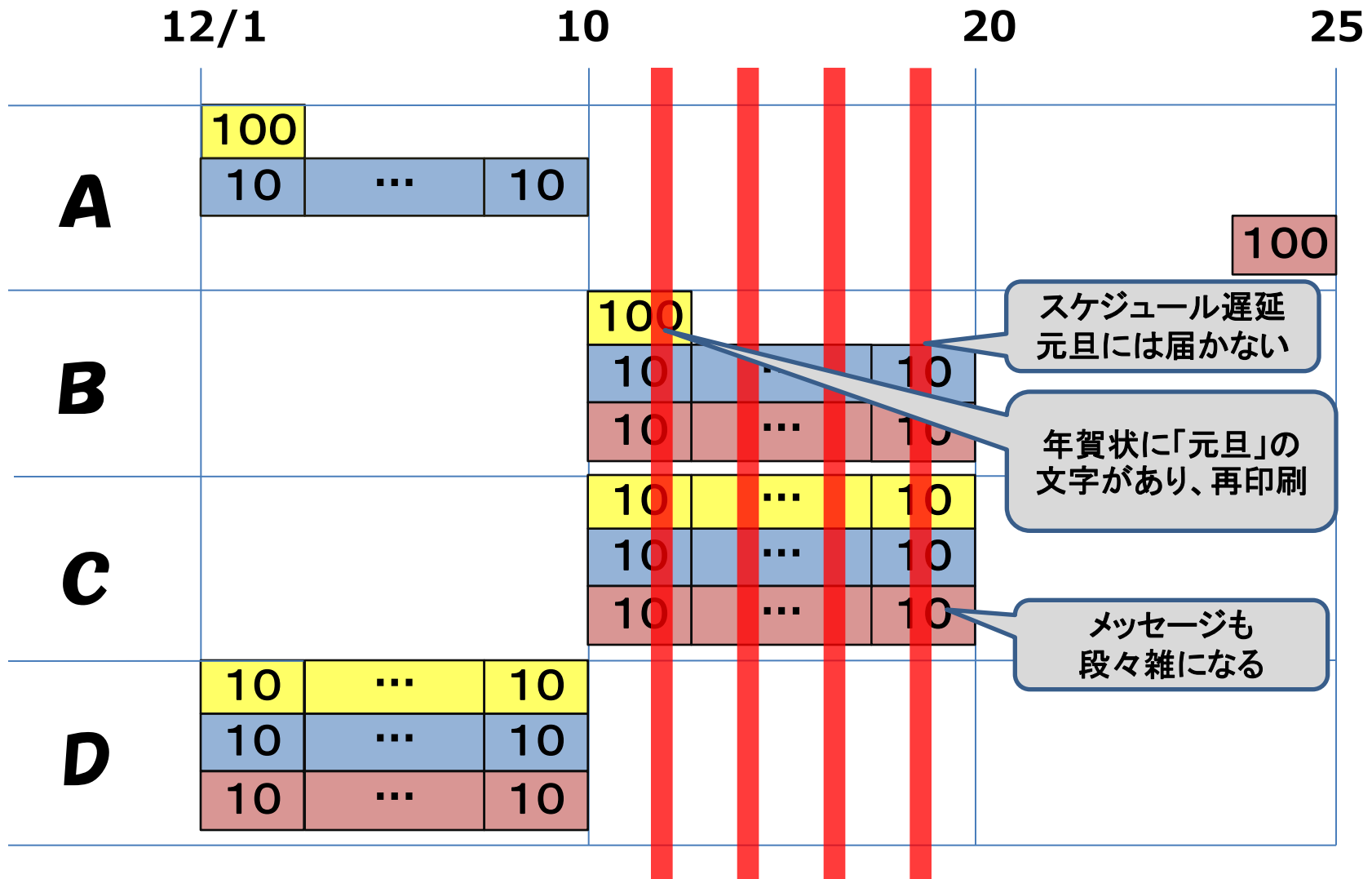
印刷済み年賀状にジュースをこぼされる！

	12/1	10	20	25
A	100 10	...	10	100
B		0 0	...	10
C		10	...	10
D	10	...	10	

最大100枚
やり直し

投函済み
無事

忘年会の連続で全く作業できず！



鞆を電車に忘れ、年賀状紛失！

	12/1	10	20	25
A	100			
	10	...	10	
B		100		
		10	...	10
		10	...	10
C		10	...	10
		10	...	10
		10	...	10
D	10	...	10	
	10	...	10	
	10	...	10	

✕切に間に合わず

振り返り

	12/1	10	20	25
A	100 10 ... 10	在庫のムダ		100
B	効率的 不確実性に弱い	100 10 ... 10 10 ... 10		
C	Aより非効率 不確実性が発生時の ムダを最小限に抑え られる	10 ... 10 10 ... 10 10 ... 10		
D	10 ... 10 10 ... 10 10 ... 10		スケジュール後半の トラブルは、納期に影響	

効率優先でなく 不確実性を考慮



タスク中心のマネジメント



タスクを細かくすればするほど
見積もり制度があがる

全てを細かく管理すれば
プロジェクトはうまくいく

タスク中心のマネジメント



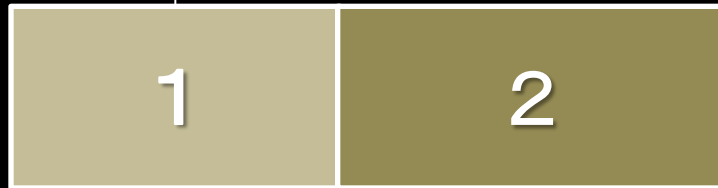
“不確実性が高い業務では
タスク中心マネジメントは
うまくいかない”



タスク中心から
不確実性を考慮した**バッファ**中心のマネジメントへ

本日の流れ

リーン
ソフトウェア開発



CCPM
スケジューリング

CCPM

スケジューリング

CCPMスケジューリング (Critical Chain Project Management)

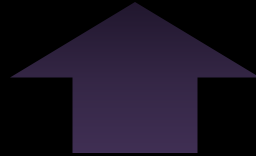


- ・プロジェクトの不確実性を考慮し、
不確実性に柔軟に対応できるプロジェクト管理手法
- ・1997年、エリヤフ・ゴールドラット博士により著書
- ・TOC(制約理論)の基本原理を適用

CCPM

スケジューリング方法

タスクに入っているバッファを取り除き、
納期前に集中配置する方法で計画する



従来のスケジューリング方法は、
4つの人間の行動特性
でスケジュールが遅れる

4つの人間の行動特性

1. 学生症候群
2. 悪いマルチタスキング
3. 早期完了の未報告
4. パーキンソンの法則

1. 学生症候群



納期間際にならないと作業を始めない行動特性

余裕を見込んだタスクでも、
タスク開始前までに全てその余裕が消費される

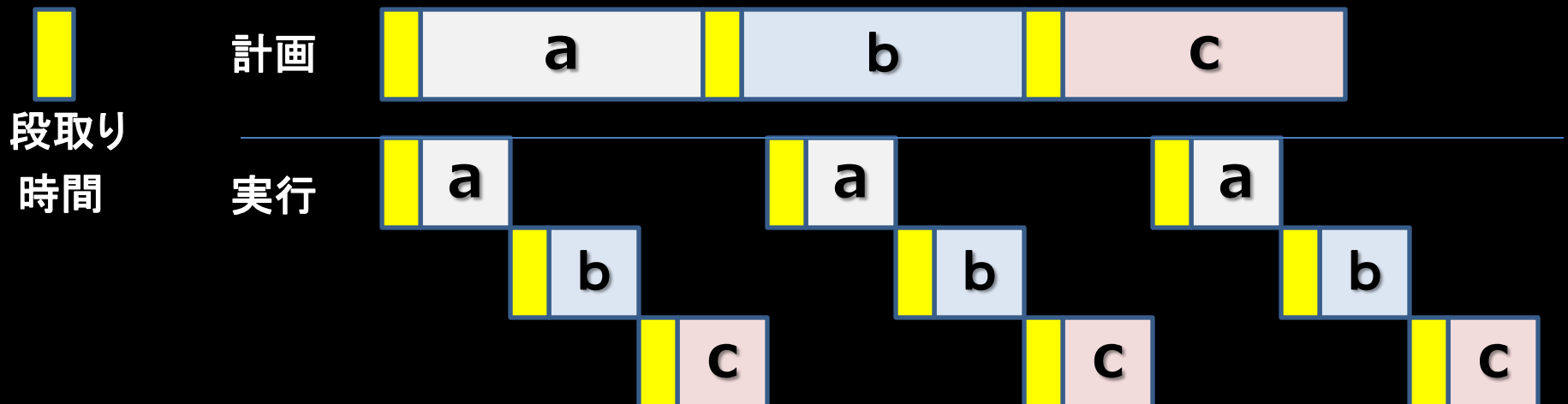
予期せぬ問題が起こったら、
バッファがないため遅れる可能性がある

人間の行動特性

2. 悪いマルチタスキング



1つのタスクが完了していないのに、
個人的な思いや感情で別のタスクにとりかかること



3. 早期完了の未報告



タスクが計画より早く終わっても報告せず、
納期になるまで次工程に引き渡さないこと

完了基準が明確でない場合、
時間が余れば見直しや、仕上げに費やされる

人間の行動特性

4. パーキンソンの法則



仕事の量は、与えられた時間を使いきるまで拡張する

(例) 頑張れば2日で終わるタスク

提出期限	2日後	7日後	10日後
10日後	学生症候群 悪いマルチタスキング	タスク 実行	早期完了 の未報告
7日後	学生症候群 悪いマルチタスキング	タスク 実行	早期完了 の未報告
2日後	タスク 実行		

CCPM

スケジューリング方法

タスクに入っているバッファを取り除き、
納期前に集中配置する方法で計画する



従来のスケジューリング方法は、
4つの人間の行動特性
でスケジュールが遅れる

バッファ中心のマネジメント



制約条件(「納期」)の直前のバッファを
中心にPDCAサイクルを回す

タスク中心のマネジメントから、
ものの見方や考え方をガラッと変えなければ、
状況を変えることは難しい

従来の手法とCCPMの手法

従来の手法

CCPM

計画立案	必要作業を今日時点から列挙 なるべくはやく開始 ムダがないよう計画	完成時点から必要作業を列挙 なるべく遅く開始 バッファを初めから計画
進捗管理	計画通り進むよう管理 遅れたら素早い対応 さまざまな管理を厳密に タスクを管理	作業は半分くらい遅れるべき 遅れても即行動しない プロジェクトを単純に管理 プロジェクト全体を管理
作業方針	作業者は効率を良くする 必要な物を必要なときに引き渡す	作業者はスピードを優先 完成したらすぐに引き渡す

バッファマネジメントの全体像

- PLAN** : バッファを計画する
- DO** : バッファの消費を記録する
- CHECK** : バッファの消費状況を確認する
- ACT** : バッファを回復させる消費状況を確認する

PLAN: バッファを計画する

1. 確保するバッファの大きさを計画する

⇒ バッファの大きさはプロジェクトの大きさをもとに決めるため、まずはプロジェクトの大きさを見積もる

2. プロジェクトの全体期間は、タスクの所要日数を50%確率で見積もる

⇒ リソース競合を排除した上で、最も長いパスの長さ（クリティカルチェーン）

⇒ クリティカルチェーンに一定比率を掛けたものがバッファの大きさになる

DO: バッファの消費を記録する

1. 計画に従ってタスクを実行する

⇒担当者は、タスクを受け取ったら全速力で完了させ、次のタスク担当者に渡すことを意味する「リレー走者の原理」に従って実行する

2. 進捗は残日数で管理する

⇒タスクの進捗状況は、あと何日でそのタスクが完了するかを表す「残日数」で収集する

⇒この残日数をもとにバッファ消費の消費状況を記録する

CHECK: バッファの消費状況を確認する

1. バッファ傾向グラフで状況確認

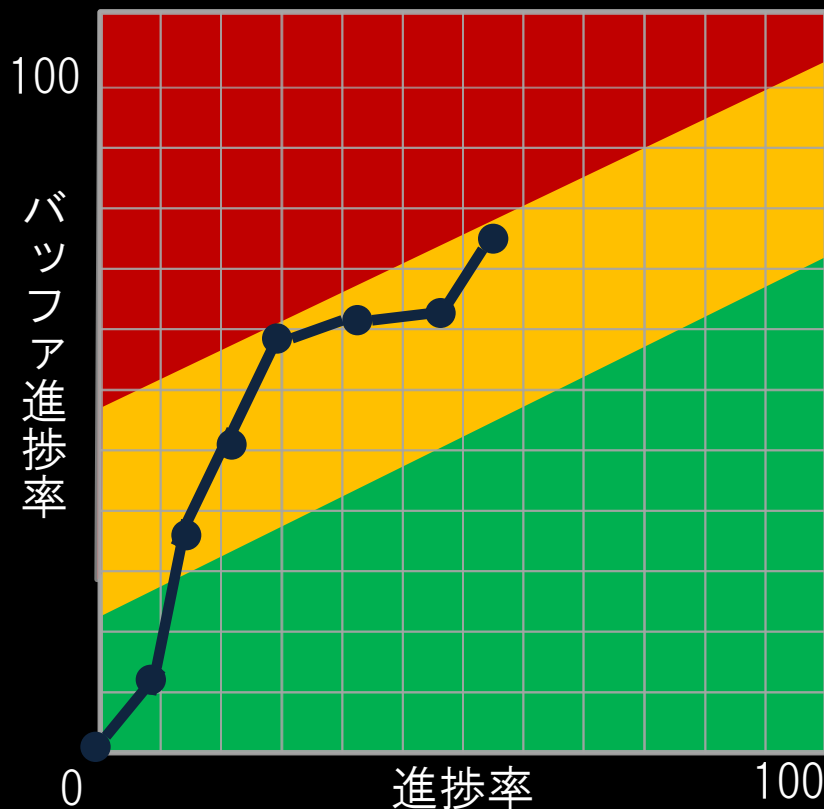
⇒横軸にプロジェクトの進捗率、縦軸にバッファ消費率

⇒バッファ消費状況によって、3つのゾーンに分かれる

緑ゾーン：安全(介入しない)

黄ゾーン：注意(対策立案)

赤ゾーン：危険(対策実行)



ACT: バッファを回復させる対策を実行する

- ・プロジェクトが赤ゾーンの場合、対策を実行する

⇒プロジェクトリーダーにはあまり権限がないため、リソース調整のような複数のプロジェクトにまたがる対策はなかなか打てない。

対策を実施する際は、組織的に行う必要がある。

スケジュール改善のカギ



1. システムを理解する

2. 遅れの原因を把握する

-偶発原因による遅れ

-異常原因による遅れ

偶発原因による遅れ



1. 見積もりの誤差による軽微な遅れ
2. 異常原因をすべて取り除いても残る

⇒改善するには

- ・方針上の問題除去など管理者が改善検討しなければならない

異常原因による遅れ



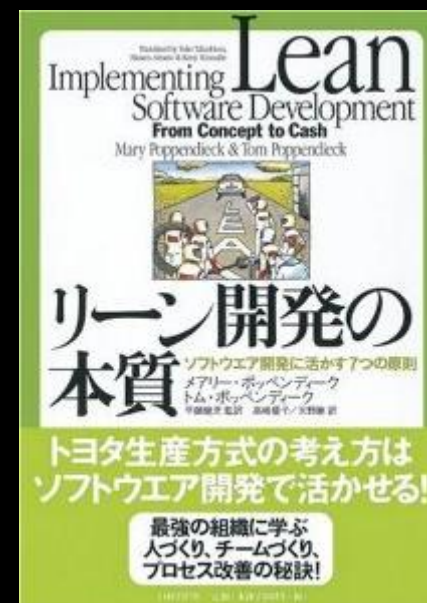
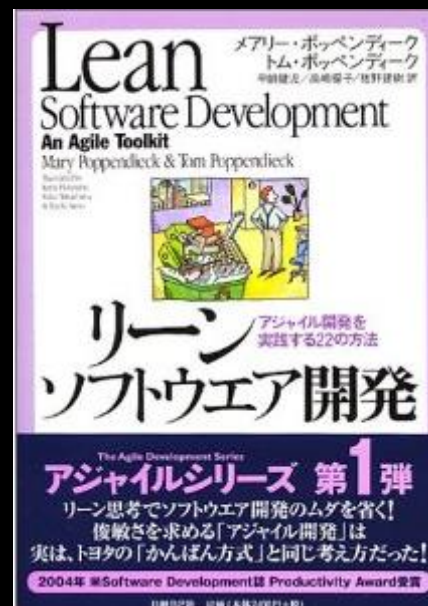
1. 特定の問題によって遅れが発生しているもの
2. 原因が明確に特定できる
3. 特定のリソースの異常、エンジニアのミスなどにより発生する

⇒管理者は、偶発原因と異常原因を理解して、バッファ管理を行うことが重要

リーン開発に 学ぶCCCPM の改善点

リーン思考の7原則

1. ムダをなくす
2. 品質を作り込む
3. 知識を作り出す
4. 決定を遅らせる
5. 速く提供する
6. 人を尊重する
7. 全体を最適化する



従来の手法とCCPMの手法

従来の手法

CCPM

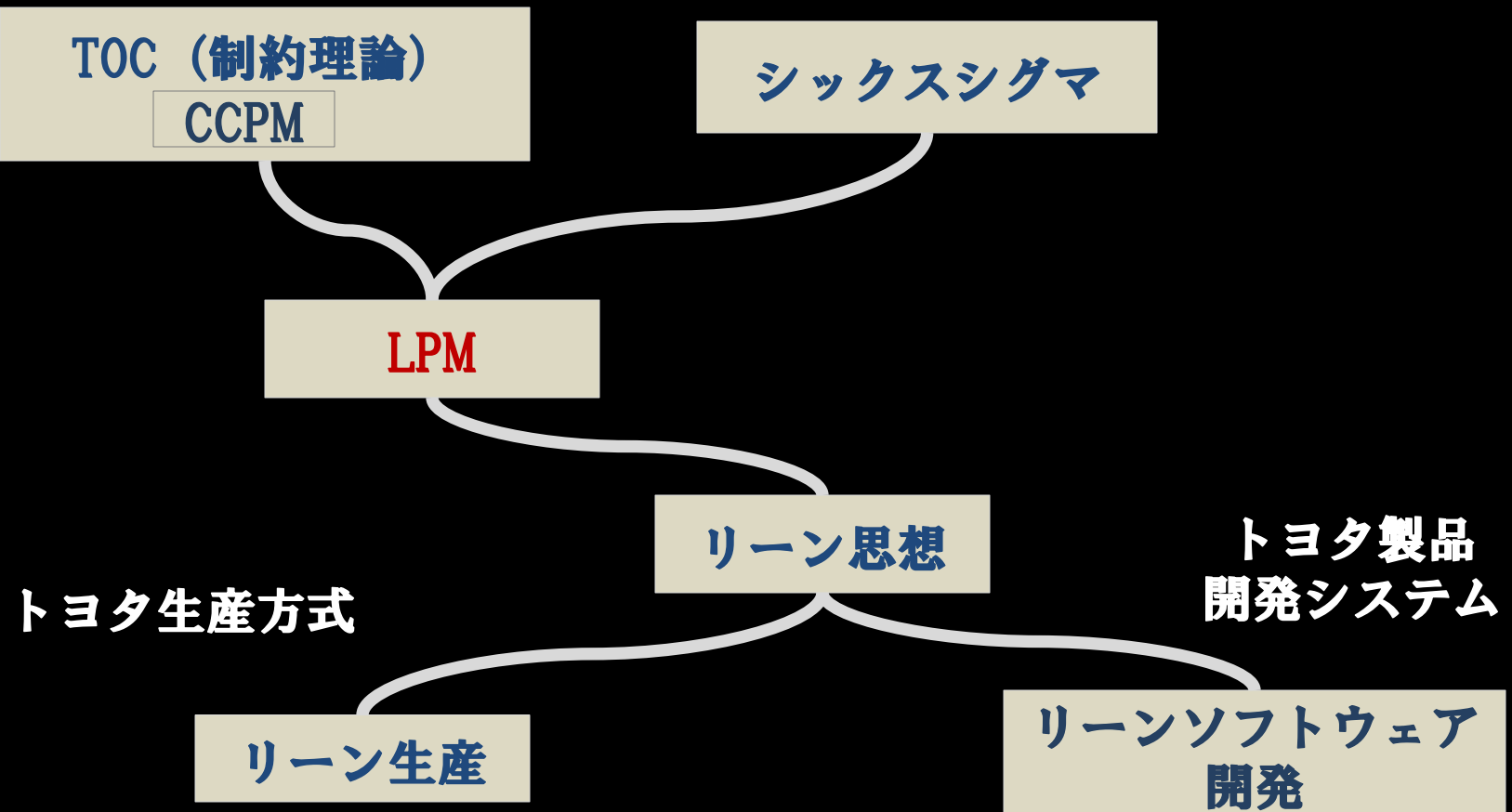
計画立案	必要作業を今日時点から列挙 なるべくはやく開始 ムダがないよう計画	完成時点から必要作業を列挙 なるべく遅く開始 バッファを初めから計画
進捗管理	計画通り進むよう管理 遅れたら素早い対応 さまざまな管理を厳密に タスクを管理	作業は半分くらい遅れるべき 遅れても即行動しない プロジェクトを単純に管理 プロジェクト全体を管理
作業方針	作業者は効率を良くする 必要な物を必要なときに引き渡す	作業者はスピードを優先 完成したらすぐに引き渡す

まとめ

1. CCPMを採用することで、不確実性に踊らされない、スケジューリングが行えるようになる
2. CCPMは、リーン開発の「ムダ」を省く考え、「アジャイル」で開発する考えを取り入れることでより効果を発揮する
3. ウォータフォール開発の現場ではCCPMを取り入れるところから始め、次のステップとしてリーン開発を取り入れることを提案する

LPM

～リーンプログジェクトマネジメント～



Q&A

ご清聴ありがとうございました