SingleProjectへの CCPM導入・運用プロセス

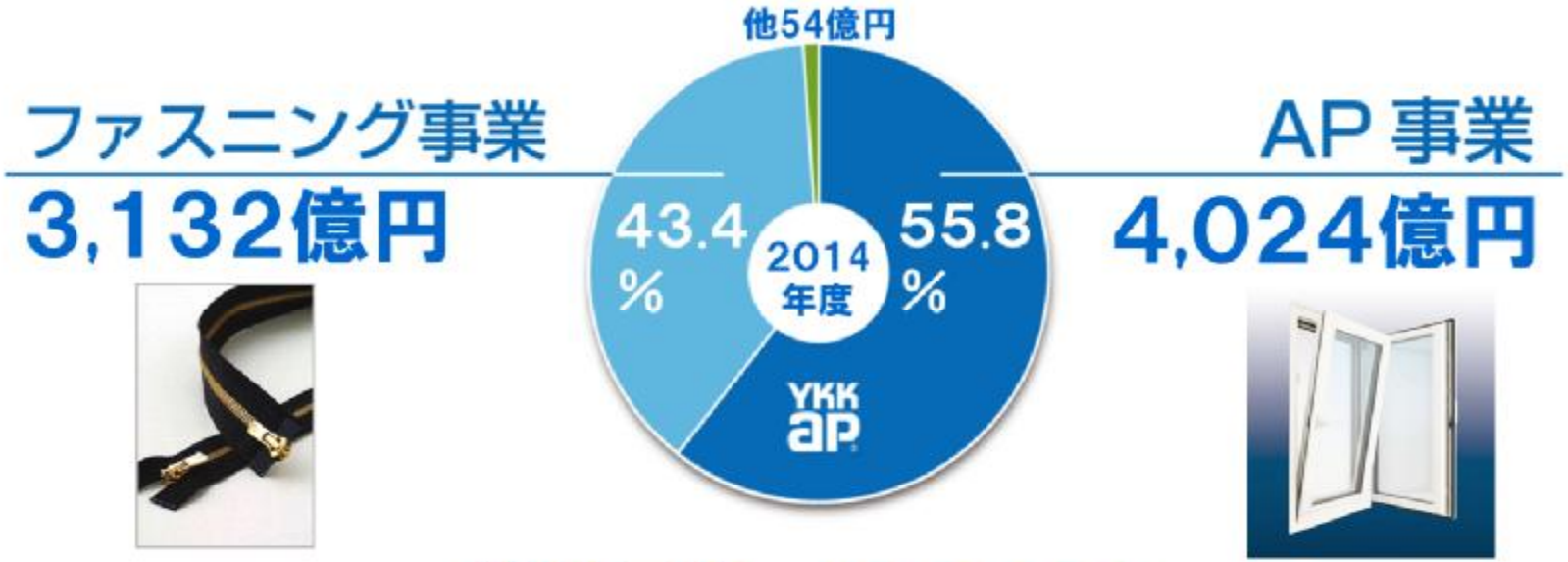
2016年1月27日
ＹＫＫ ＡＰ株式会社
情報システム部
岩宗　幸弘

目次

1.会社プロフィール

YKKグループ 連結売上高

連結売上高　7,210億円　※2014年度実績



従業員数　42,000名　※2015年3月末

（国内17,200／海外24,800）

1.会社プロフィール

YKKグループ 事業拠点

世界71カ国／地域　111社（585拠点）

◎国内24社（283拠点）

◎海外87社（302拠点）

北中米
USA、カナダ、メキシコ、ホンジュラス、コスタリカ、
エル・サルバドル、トリニダード・トバゴ、グアテマラ

南　米
ブラジル、チリ、アルゼンチン

アジア・オセアニア
中国、香港、マカオ、韓国、台湾、シンガポール、タイ、ベトナム、
フィリピン、マレーシア、インドネシア、カンボジア、インド、
バングラデシュ、スリランカ、パキスタン、オーストラリア、
フィジー、ニュージーランド

ヨーロッパ
オランダ、英国、ロシア、ノルウェー、スウェーデン、フィンランド、デンマーク、
リトアニア、アイルランド、ドイツ、ポーランド、ウクライナ、ベルギー、フランス、
チェコ、オーストリア、ハンガリー、ルーマニア、スイス、クロアチア、トルコ、
セルビアモンテネグロ、ブルガリア、イタリア、スペイン、ポルトガル、ギリシャ

アフリカ
エジプト、ケニア、モーリシャス、スワジランド、南アフリカ、ベラルーシ

中東アジア
アラブ首長国、モロッコ、チュニジア、レバノン、ヨルダン

※2015年4月

1.会社プロフィール

YKK APとは、「YKK」のグループ会社で、建材製品（Architectural Products）を扱っています。

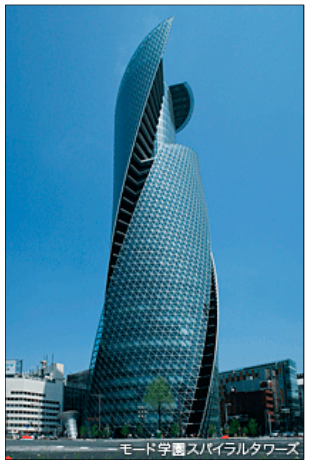
▼ 住宅建材

窓、サッシ、シャッター、雨戸、網戸、面格子、住宅用電装商品、玄関ドア・引戸、勝手口ドア、浴室出入口、室内ドア・引戸、室内階段、フローリング、バルコニー、テラス、オーニング、門扉、フェンス、カーポート、外装材、複層ガラス 等




▼ ビル建材

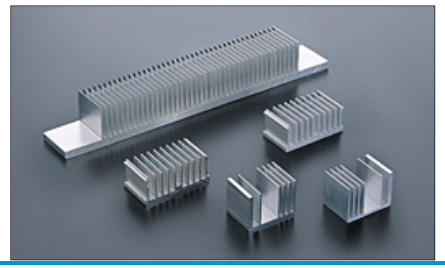
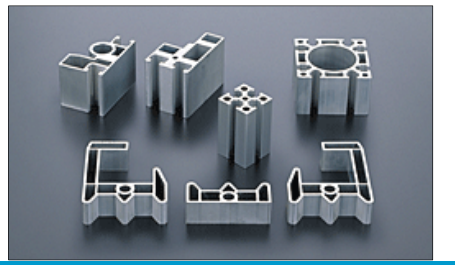
超高層・高層・中層・低層ビル用窓・サッシ・ドア・カーテンウォール、スチール商品、改装用商品、エントランス商品、自動ドア 等





▼ その他

機械部品、フレーム、ヒートシンク、建築部品 等

1.会社プロフィール（プロジェクト）

プロジェクト概要

- ü スコープ　　：生産管理業務の再構築プロジェクト
- ü ソリューション：パッケージとスクラッチ開発の組み合わせ
- ü 開発分担　　：パッケージ部分は外部ベンダに開発委託
  スクラッチ部分は自社開発

CCPM適用方針

- ü システム開発の詳細設計～システム内結合テスト　工程に適用（約６ヶ月間）
- ü 外部ベンダにも協力を依頼
- ü まずはシステム構築に絞った適用で効果確認
- ü 富士通システムズ・ウエスト様にはCCPM適用を前提としたPM支援を依頼

2.何故変えるのか

CCPM導入の背景

大規模システム構築で**QCD**を守るために、ソリューションの工夫・人員の確保だけでなく、管理面でも課題クリアが必要

●大型のシステム構築プロジェクトで顕著となる課題

- 移行タイミングは厳ン厳イントなので、スケジュールは厳守
- 複数のサブシステムが存在し、開発遅延が生じると後続工程への影響が大きくなる
- 開発は自社担当部分とベンダ担当部分があり、自社の遅れはベンダへのコスト増に繋がる

●従来からの課題

- **WBS**管理はしているが、個別タスクの進み/遅れが全体に及ぼす影響が見えず、いつの間にか遅延している
- 人は余らせていないのに、進捗が思い通りにならない
- イナズマ線では、どの位遅れているか把握できない

3.何を変えるのか？（従来の状態）

ヒアリングを通して確認された問題症状

- 進行中パス/タスクが「突発業務」によって遮断される
  （突発業務のほとんどは、課題解決ミーティング(数回/日)）
- パス/タスクのほとんどは予定より長くかかる
  （後の工程になればなるほど…）
- 必要な時に必要なリソースが使えないことが多い
- ぎりぎりで仕様が変更になって修正に追われる
- 設計中、テストでのやり直しが多すぎる
- 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
- 優先順位がころころ変わる
- 残業が多すぎる

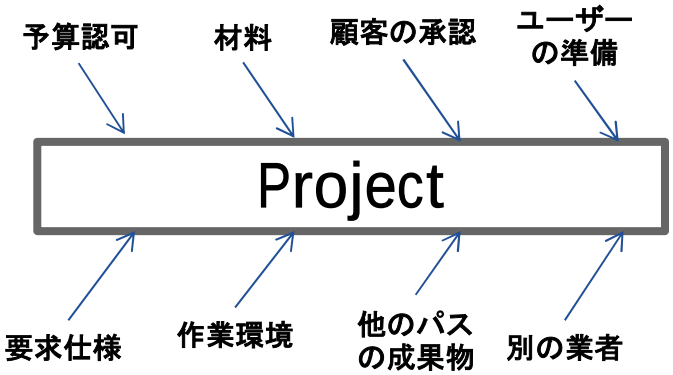
解決の糸口（CCPMの考え方）

ü タスクに要する時間は、「Work Time」と「Wait Time」に分けられる
ü CCPMでは「Wait Time」を徹底的に活用する

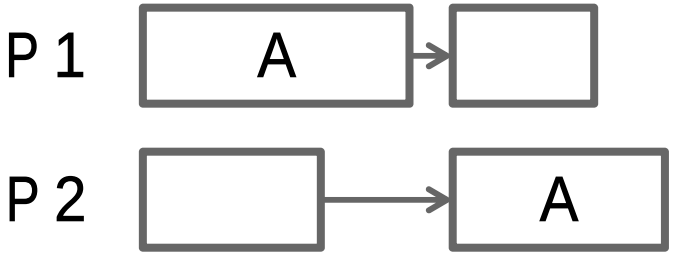
3.何を変えるのか？（従来の状態）

待ち時間が発生する４つの主な要素

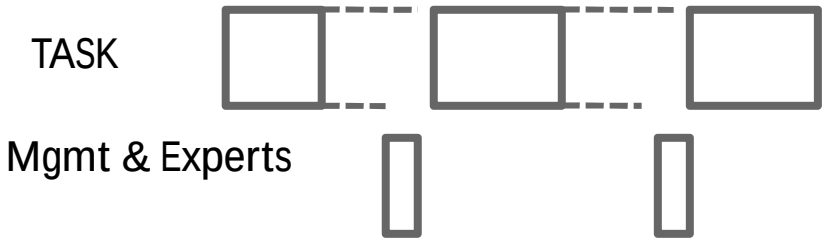
インプットが足りない



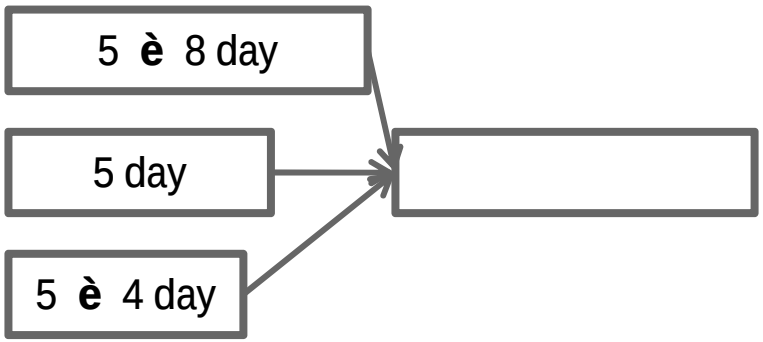
リソースが使えない
（別の仕事で忙しい）



管理者/専門家の支援が得られない
（課題解決が遅い）



合流経路が伸びる
（統合地点での非同期）



3.何を変えるのか？（従来の状態）

４要素の発生原因分析

インプットが足りない

・ 仕様提示遅延
 - ユーザの意思決定待ち
 - ユーザからの業務要件提示待ち
 - 外部IF項目提示待ち
・外部サポートからの回答待ち
・仕様変更 (詳細は仕様提示待ちと同じ)
・仕様再確認(曖昧・手戻り・不整合)
・課題対応待ち
 - 業務運用
 - システム制約
・前工程の仕様未確定事項による手直し
・環境構築待ち

リソースが使えない
（別の仕事で忙しい）

・同時進行のパス/タスクが多く発生し、マルチタスクが発生
・主力メンバの高負荷状態により、課題が解消されず、
 その間、別のタスクを開始する or 見切りでタスクを進める
・リソースがタスクに薄く広く引き伸ばされた割当計画

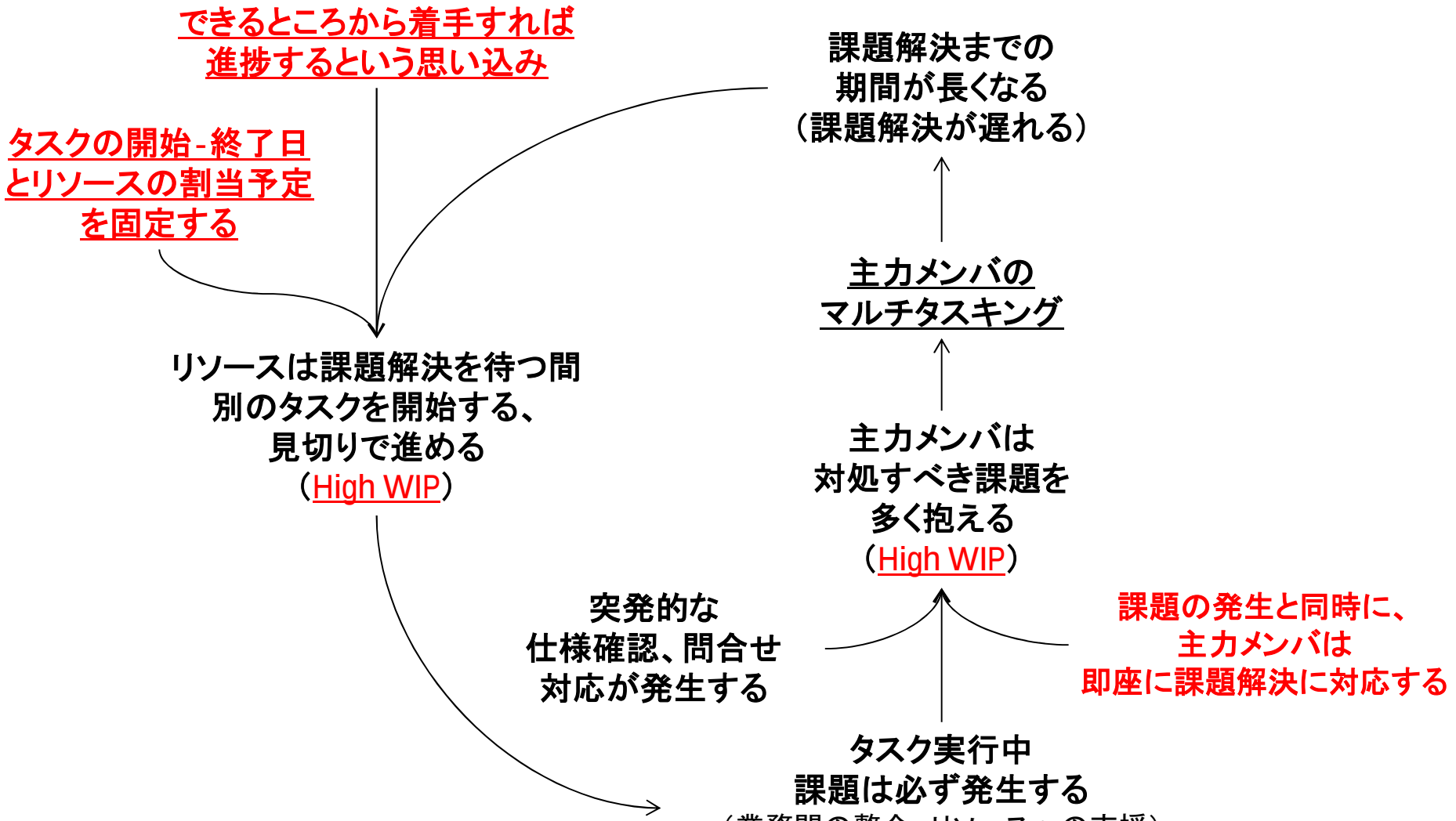
管理者/専門家の支援が得られない
（課題解決が遅い）

・主力メンバの管轄するパス/タスクが同時に
 多く発生し、目が行き届かない
・主力メンバの管轄するパス/タスクが同時に
 多く発生し、課題解決待ちの項目が増大する

合流経路が伸びる
（統合地点での非同期）

3.何を変えるのか？（従来の状態）

悪循環のサイクル（主力メンバがなぜ過負荷になるか）



WIP – Work in Process 進行中の作業

3.何を変えるのか？（問題点を纏めると・・・）

§多すぎる進行中作業

- ü 進行中パス/タスク数が多すぎる
- ü 高負荷状態の中心メンバ
- ü 薄く伸ばしたリソース割り当て

§スケジュール基準でのタスク開始

- ü 予定した開始日がくれば、準備が不十分でもそのタスクを開始する

§できるところから進捗を見せなければならない

- ü リソースの手空きは許容できない
- ü 業務単位で優先順位が局所的に決められる

計画策定の考え方と運用方法を変える

4.何に変えるのか？

- ü 多すぎる進行中作業
- ü スケジュール基準でのタスク開始
- ü できるところから進捗を見せなければならないという意識

§進行中作業の積極的な調整

パイプライニング

- §進行中パス/タスク数を制限する
- §主力メンバ(以後、ﾀｽｸﾏﾈｰｼﾞｬと呼ぶ)の集中
- §リソースの集中

§準備状況を判断してのタスク開始

フルキット

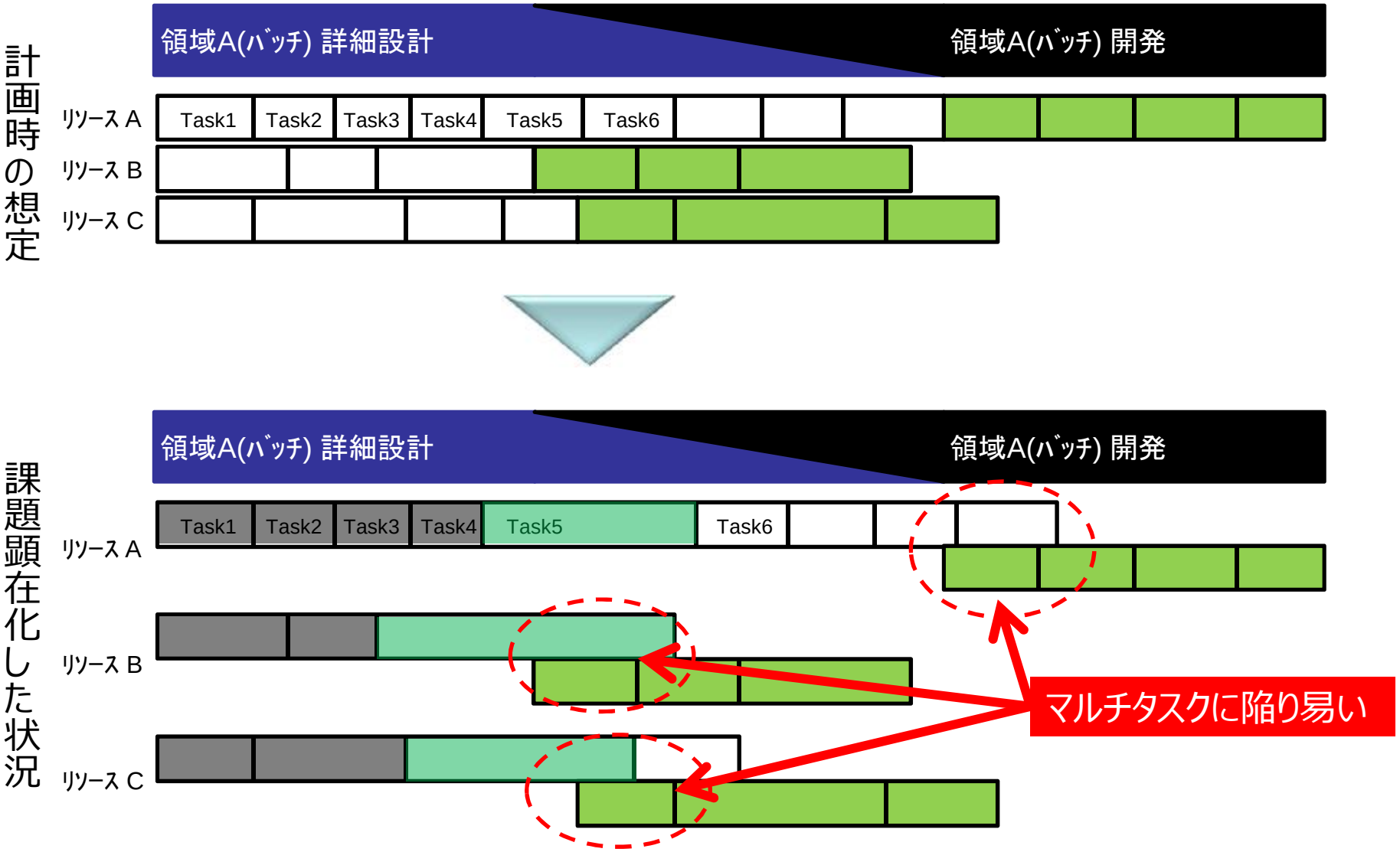
- §開始日が来ても、準備が出来ていなければタスクを開始しない

§バッファの優先順位に基づいてリソースを配置する

バッファマネジメント

- §余剰キャパシティはムダではない
- §ﾌﾟﾛｼﾞｪｸﾄ全体で共通の優先順位

計画立案　プロセス


計画時の想定
領域A(ﾊﾞｯﾁ) 詳細設計
領域A(ﾊﾞｯﾁ) 開発
ﾘｿｰｽ A
ﾘｿｰｽ B
ﾘｿｰｽ C
Task1
Task2
Task3
Task4
Task5
Task6
課題顕在化した状況
領域A(ﾊﾞｯﾁ) 詳細設計
領域A(ﾊﾞｯﾁ) 開発
ﾘｿｰｽ A
ﾘｿｰｽ B
ﾘｿｰｽ C
Task1
Task2
Task3
Task4
Task5
Task6
マルチタスクに陥り易い

5.どの様に変えるのか？ （計画立案）

Rule 1: WIP（進行中作業）を減らし、ﾒﾝﾊﾞを集中（増員）する

現状のWBSを確認し
ﾀｽｸﾏﾈｼﾞｬｰを基点に
フローを整理する

↓

リソース集中した際の
リソース数と期間を
整理する

↓

ﾀｽｸﾏﾈｼﾞｬｰを
必要に応じて追加する

↓

ﾀｽｸﾏﾈｼﾞｬｰを基点に
Low WIP Plan
を作成する

↓

現実的なﾘｿｰｽ数を基に
Low WIP Planを
修正する

増員＆期間短縮

A領域

		Original Plan			Low WIP (リソース集中) Plan		
機能	フェーズ	ﾀｽｸﾏﾈｼﾞｬｰ（TM）	Phase CT	リソース数	ﾀｽｸﾏﾈｼﾞｬｰ（TM）	Phase CT	リソース数
A領域（画面）	詳細設計	甲さん	21	1 人	甲さん	15	2 人
	開発	甲さん	39	1 人	甲さん	30	3 人
	テスト	甲さん	20	3 人	甲さん	15	4 人
A領域（バッチ）	詳細設計	乙さん	31	3 人	甲さん+乙さん	12	7 人
	開発	乙さん	39	4 人	甲さん+乙さん	20	8 人
	テスト	乙さん	20	4 人	甲さん+乙さん+丙さん	8	13 人

D領域

		Original Plan			Low WIP 計画		
機能	フェーズ	ﾀｽｸﾏﾈｼﾞｬｰ（TM）	Phase CT	リソース数	ﾀｽｸﾏﾈｼﾞｬｰ（TM）	Phase CT	リソース数
D領域（画面）	内部設計・開発	Aさん	26	0.5 人	Aさん	12	1.5 人
	外部設計	Aさん	51	1.0 人	Aさん	30	2.5 人
	内部設計・開発	Aさん	54	2.0 人	Aさん	36	3.0 人
	内部設計・開発	Aさん	29	1.0 人	Aさん	14	3.0 人
	外部設計	Aさん	32	3.0 人	Aさん	27	4.5 人
	外部設計	Aさん	25	0.5 人	Aさん	13	1.5 人

5.どの様に変えるのか？ （計画立案）

Rule 1: WIP（進行中作業）を減らし、ﾒﾝﾊﾞを集中（増員）する

§ **進行中作業の積極的な調整**
- ü 進行中パス/タスク数を制限する
- ü 主力メンバ(以後、ﾀｽｸﾏﾈｰｼﾞｬと呼ぶ)の集中
- ü リソースの集中

パイプライニング

§ **準備状況を判断してのタスク開始**
- ü 開始日が来ても、準備が出来ていなければタスクを開始しない

§ **バッファの優先順位に基づいてリソースを配置する**
- ü 余剰キャパシティはムダではない
- ü ﾌﾟﾛｼﾞｪｸﾄ全体で共通の優先順位

まとめ		ﾀｽｸﾏﾈｰｼﾞｬが同時に管轄するタスク数	ﾀｽｸﾏﾈｰｼﾞｬが同時に管轄するフェーズ(工程)数	ﾀｽｸﾏﾈｰｼﾞｬが同時に管轄するﾌｪｰｽﾞ当りﾀｽｸ数
YKK AP担当分	現行（High WIP Plan）	10件程度 / TM	1-2フェーズ/TM	3-8件程度 / 1ﾌｪｰｽﾞ
	Low WIP Plan	5件程度 / TM	1フェーズ/TM	5件程度 / 1ﾌｪｰｽﾞ

Rule 3: 50%のバッファを持った計画を立てる

① 現行の計画（オリジナル）

作業期間＝7 weeks (35d)/メンバ 3人

② ﾒﾝﾊﾞ集中による計画

管理者が支援を実行することで
得られる短縮
（素早い課題解決＆手直し軽減）

作業期間=3.5 weeks (18d)
/メンバ 6人

③ 50％期間カット&50％のバッファ

【カット期間算出】
②×50% = 1.75weeks(≒8d)
【バッファ期間算出】
1.75weeks×50% = 0.875(≒5d)
【計画上の最長実施期間】
＝カット期間残(②の50%)
＋バッファ期間(②の50%×50%＝25%)

※タスクの日数を半分にする目的は、バッファの消費状況でプロジェクトのリスクを早く洗い出すため。（半分の日数で実行するためではない）

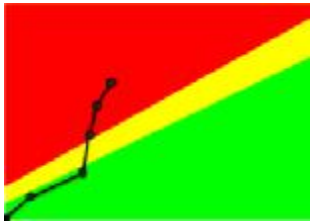
5.どの様に変えるのか？（実行管理）

●ツールを使い分けて管理を実施

ü サブシステム／工程　　毎の残日数　　⇒**CONCERTO**

ü サブシステム内のタスク　毎の残日数　　⇒**Excel**

●**CONCERTO**で管理したいこと



ü 全体のクリティカルチェーン確認

ü リソースを必要としてる工程確認

●**CONCERTO**での管理では不便なこと

ü 工程内の**WBS**作成

ü 細かい粒度でのタスク管理

ü タスクの追加/削除

ü タスクの担当者変更

ü **Excel**と比較した場合の親和性・コスト

5.どの様に変えるのか？（実行管理）

●ツールを使い分けての管理

ü サブシステム／工程　　毎の残日数　　⇒**CONCERTO**

ü サブシステム内のタスク　毎の残日数　　⇒**Excel**

●**CONCERTO**での管理単位

※**TM**＝ﾀｽｸﾏﾈｰｼﾞｬ

	TMアサイン単位	CONCERTO登録工程		
領域**A**	バッチ系機能	詳細設計	プログラミング	テスト
	画面系機能	詳細設計	プログラミング	テスト
領域**B**	バッチ系機能	詳細設計	プログラミング	テスト
	画面系機能	詳細設計	プログラミング	テスト
領域**C**	バッチ系機能	詳細設計	プログラミング	テスト
	画面系機能	詳細設計	プログラミング	テスト

●**Excel**での管理単位

機能単位（工程毎）

5.どの様に変えるのか？（実行管理）

タスク更新ミーティング – メンバ残日数 申告方法

計画日数

- ｲﾝﾌﾟｯﾄが揃っていないかもしれない
- 手直しが必要になるかもしれない
- 実施中に分からないことがあるかもしれない
- 管理者が忙しすぎて、対応してくれないかもしれない

WBS 当初日数 **4日**

CCPM 計画日数 **2日** → 1日 ﾊﾞｯﾌｧ

進行中 残日数の申告方法

- ｲﾝﾌﾟｯﾄが揃っていないかもしれない
- 手直しが必要になるかもしれない
- 実施中に分からないことがあるかもしれない
- 管理者が忙しすぎて、対応してくれないかもしれない

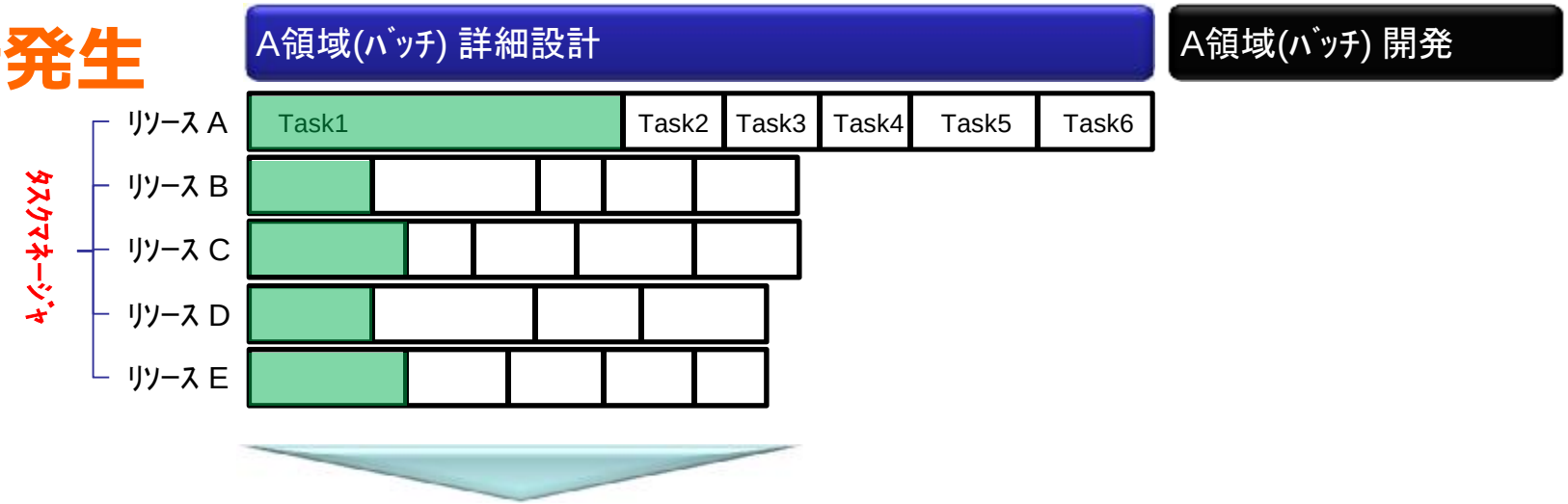
CCPM 計画日数 ~~2日~~ **1.5日**

タスクマネージャが翌日以降、上記事項を即座に解消する（停滞することがない）とした場合の日数を残日数として、申告する。
但し、当日、即座に対応してくれなかったことで発生した遅れに関しては、支援依頼として申告する

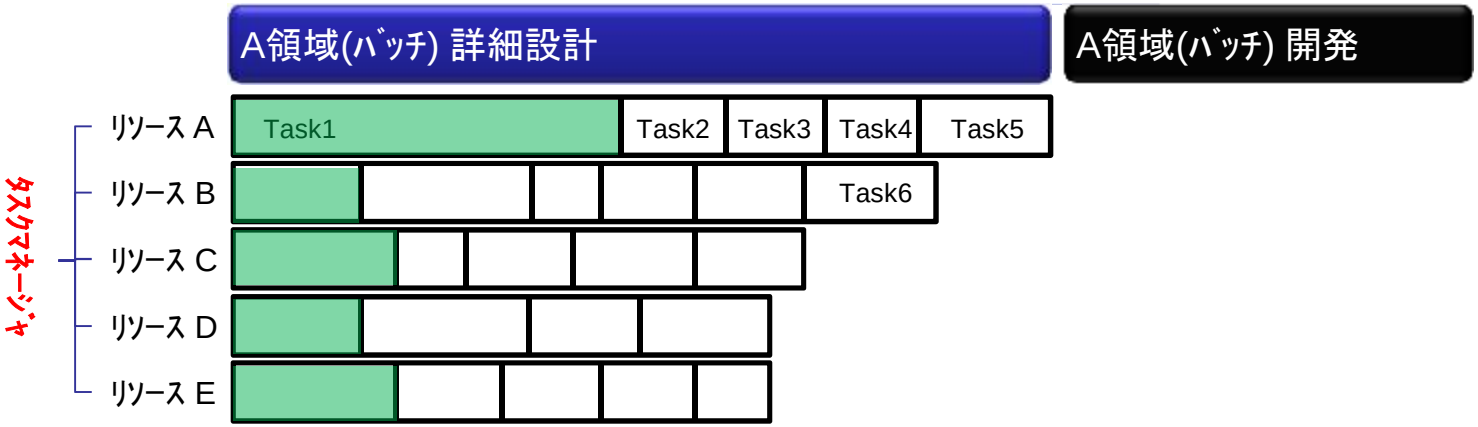
5.どの様に変えるのか？（実行管理）

日々の残日数調整

バラつき発生



平準化



進捗に伴い変化するメンバの残日数を調整する

（フェーズ全体を短縮するよう、 リソースの割当を柔軟に行う）

残日数の把握・更新 プロセス

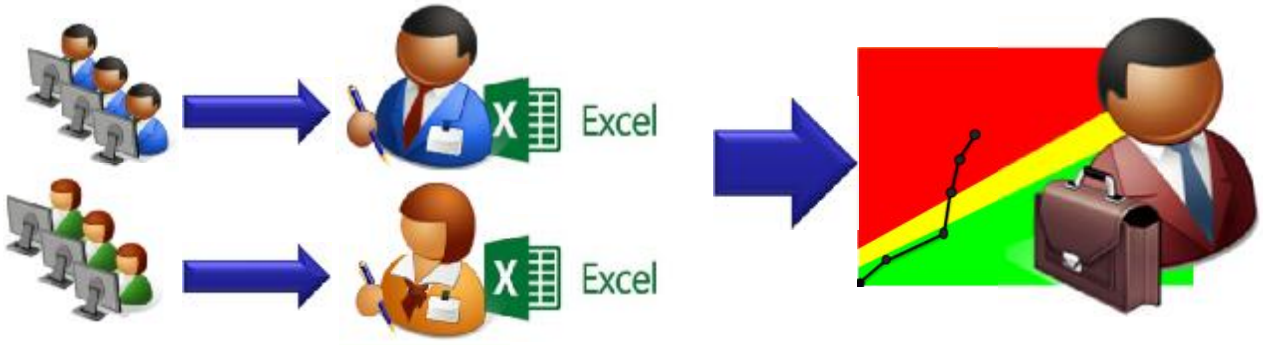
●ツールの使い分け

ü サブシステム／工程　毎の残日数　⇒**CONCERTO**

ü サブシステム内のタスク　毎の残日数　⇒**Excel**

●プロセス

1. サブシステム内の各タスクの残日数確認、**Excel**入力【メンバ⇒ﾀｽｸﾏﾈｰｼﾞｬ】
2. **Excel**でサブシステム／工程単位の残日数算出
3. チーム内ミーティングでリーダに報告【ﾀｽｸﾏﾈｰｼﾞｬ⇒リーダ】
4. リーダは確認後、**CONCERTO**へ残日数を反映しバッファ確認



5.どの様に変えるのか？ （リーダによるタスクマネージャ確認）

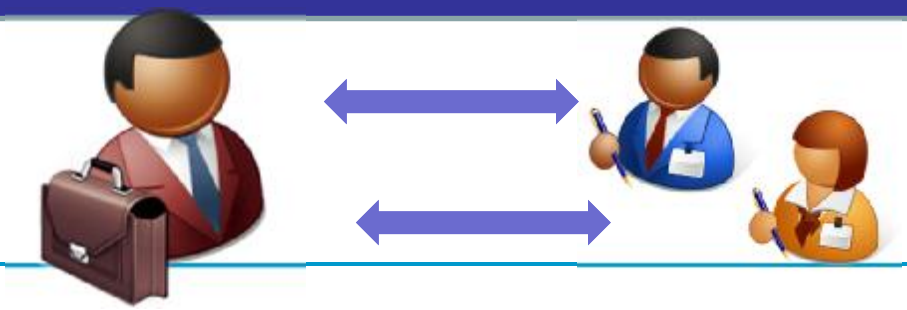
- 毎日、短時間のミーティングで状況を確認

●残日数で終る作戦が考えられているか

●バッファは適切に利用されているか

ムダ遣い	上手な使い方
• やり直し • 引き継ぎ内容に漏れがあった • 課題発見の遅れ • 中途半端な課題解決 • 準備が不完全なのに作業した • バッファの優先順位に従わない	• 技術的問題 • お客様からの変更 • 予見できなかった失敗 • スキルレベルの違い • 体調不良/忌引きによる欠勤

●ﾀｽｸﾏﾈｰｼﾞｬは手に余る課題を抱えていないか



5.どの様に変えるのか？（まとめ）

- 計画立案
  - 仕事に人を割り当てる
  - リソースを集中する
  - 同時実行タスク数はタスクマネージャが捌ける範囲

- 実行管理（毎日実行）
  - バッファの優先順位に基づくタスク分担調整
  - タスクマネージャが先回りできているかの確認
  - タスクマネージャの手に余る課題の確認

●苦労

「上位層への報告」

- 理由
  - ü 進捗率やタスク完了数での報告が主流なところに残日数やトレンドチャートといった指標をどのように理解してもらうか
- 結果
  - ü 初回報告は理屈が分からないと見事撃沈
- 対応
  - ü 残日数を導き出すプロセスと骨格となるWBSの存在を説明し理屈を理解してもらった
  - ü トレンドチャートは毎回動いた方向・幅の理由を解説し理解を深めてもらった

●気づき

「CCPM導入」

- バッファ無し納期を目指すことが重要
  - ü バッファが頭にあると知恵は絞り出されない
  - ü リーダ自身がバッファ無し納期を目指す姿勢が無いと、タスクマネージャやメンバはバッファ無し納期を目指さない
  - ü メンバの待ちを作らないようにタスクマネージャが先回りできているかをリーダは日々確認する
- 管理対象タスクの向き不向き
  - ü 管理外リソースによる実行タスクの管理には不向き。簡易な管理方法を適用した方が良い。
    - 例：ユーザトレーニングの実施、ユーザへ依頼するマスタデータ作成
- 独自知識だけでは限界がある
  - ü 富士通システムズ・ウエスト様に支援頂いたが、巷のCCPM本の知識だけでは誤った運用に至っていたと評価

●気づき

「CCPMが果たすプロジェクト管理プロセス」

- 背景
  - ü フルキットを工程へのエントリークライテリア(品質管理ﾌﾟﾛｾｽの一部）と解釈
- 結果
  - ü 工程に対するチェック項目となり、フルキット実施期間が間延びしたものとなってしまった
- 次やるとしたら
  - ü フルキットは「タスクを終らせるための準備が整っているか？」という内容で活用してみる

- 納期短縮目標

全サブシステムの開発完了（開発期間6ヶ月）
：従来手法での計画・管理より2週間前倒し完了

結果：**達成**

【内訳】

全体工期	⇒2週間短縮(約8%)
自社開発部分（直接管理）	⇒約1ヶ月短縮(約16%)

自社開発部分は、課題対応とリソース融通をタイムリーに行った結果として、リソースの待ち時間を削減できた結果と評価。

8.今後の取り組み計画

シングルプロジェクトでの適用結果
QCD達成への効果が確認できた

●開発の中心部隊への組織プロセス導入を開始

上位層からの要求
・生産性向上
　　同じリソースでアウトプットを高める
・ユーザ部門からの信頼獲得
　　スピーディーな保守対応実現
　　年間テーマの達成率向上
・リソースの有効活用
　　全社プロジェクトへリソースを融通しても揺るがない体制構築

活動継続中