



SPI Japan
ソフトウェアプロセス改善カンファレンス2025

大規模・多層構造プロジェクトにおけるマネジメントプロセス改善 — 統合管理ビューによる予兆検知とマネジメント品質向上の実践 —

2025年10月22日(水)

本報告は**INTLOOP Project Management株式会社**としての立場から、
株式会社NTTデータグループ様のプロジェクトにおける**取り組み事例**をもとに構成しております。
本内容の開示にあたっては、**両社間で事前に合意の上**、
プロジェクト情報の一部をSPIコンソーシアムにて共有させていただくものとなります。

はじめに

プログラムマネジメントで、こんな悩みをお持ちじゃないですか？

複数のチームが同時に開発を進め、維持業務も並行して走る——
そんな複雑な環境で、**進捗・品質・会議運営をどうコントロールするか。**

私たちは、「統合管理ビュー」という仕組みを導入し、

- ・進捗情報の見える化
- ・兆候の早期検知
- ・会議運営の質的向上

を実現しました。

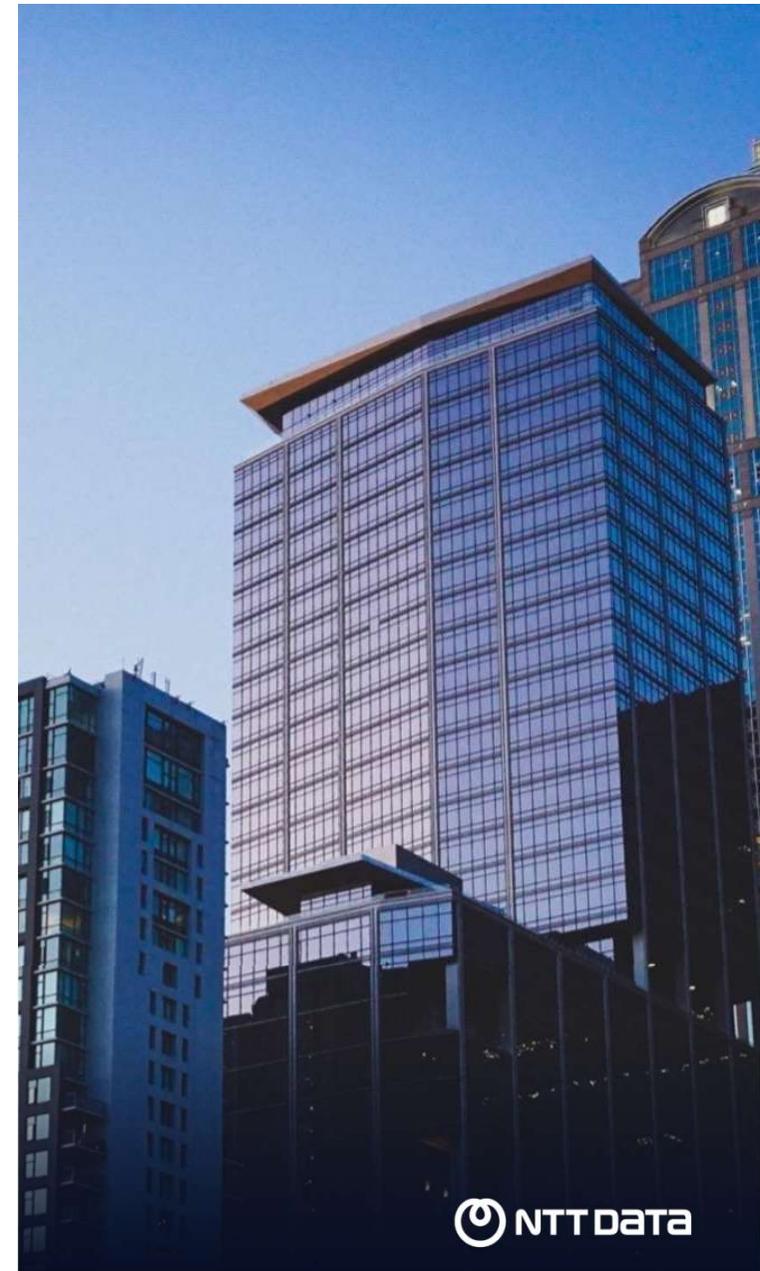
本日のプレゼンでは、以下の流れでお話します。

1. プログラムマネジメントの難しさ
2. 当プロジェクトの課題と解決策
3. 実際の改善効果
4. リーダー層の声と妥当性の検証
5. 振り返りと今後の展望

1. プログラムマネジメントの難しさ

2. 当プロジェクトの課題と解決策
3. 実際の改善効果
4. リーダー層の声と妥当性の検証
5. 振り返りと今後の展望

マネジメントの難しさを整理します。



プロジェクトを進める中で、こんな悩みはありませんか？

- ◆進捗が思うように進まない
- ◆課題がどこにあるのか分からない
- ◆チーム間の連携がうまくいかない
- ◆トラブルが起きてからしか気づけない

→課題を、2つの構造的な要因に分類

1. 立場の違いによって見える範囲が変わる

役割ごとに視点が異なり、課題の捉え方や気づき方にズレが生じる

2. 開発と維持保守が並走するケースが増えている

異なる性質の業務が組織内・チーム内で同時進行し、マネジメントの難易度が上がっている

上記2つの課題により、進捗・品質・課題などを横断的に捉える“予兆”が見えづらく、後続リスクの判断が困難となる。

プロジェクト推進は課題が多いけど、結局何が大事なんだろう。



難しいね。
自分のことで精いっぱいだよ。
情報を整理しないと。

立場の違い

各層の視座の違いにより、予兆の捉え方にもズレが生まれる。
そのズレが、トラブルの早期発見や共有を難しくしている。

問題はどこにあるのか、全体を把握したいな。

※PM：プロジェクトマネージャー PL：プロジェクトリーダー



■ トップ層目線 (PM・PL)

全体の把握と判断に苦勞しており、予兆を拾う仕組みが不足している。

他のチームとも連携しなきゃ。

※TL：チームリーダー STL：サブチームリーダー



■ リーダー層目線 (TL・STL)

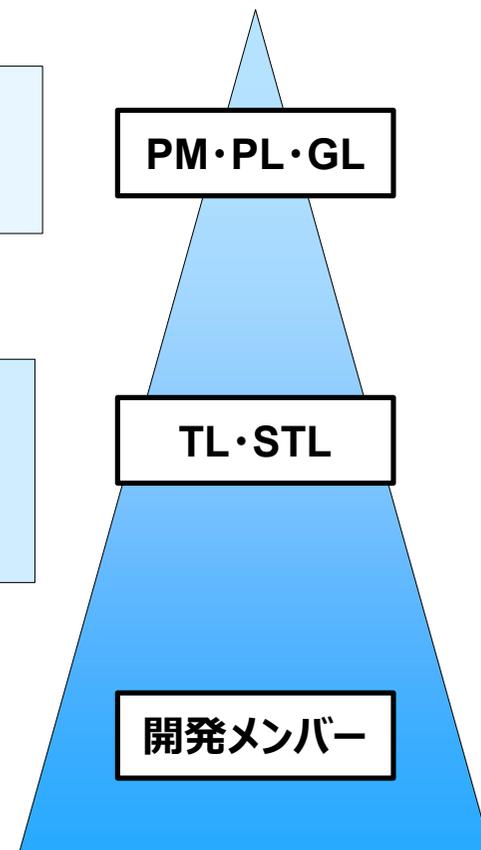
チーム単独での対応に限界があるが、他チームとの連携や支援が難しい。
進捗と計画がずれやすく、予兆の判断・共有が難しい。

このタスクって、そんなに重要なんだ。



■ 開発メンバー層

自分の役割や影響が見えず、予兆に気づいても動きづらい



参考：各組織層の注視領域

それぞれの立場で注視すべき領域が異なっており、リーダー層以下は視野が限定されやすい。特定の管理領域に集中するあまり、兆候や変化を見落とし、トップ層への報告が漏れてしまう。
 → 進捗管理を進める上で、役割を越えた横断的な視点と情報の橋渡しが不可欠。

視座\管理領域	進捗管理・予兆管理	課題管理	リスク管理	品質管理	ステークホルダー調整	影響先
トップ層 広い視野 (全体俯瞰)	各リーダー層からの報告をベースに意思決定を行う →スケジュール変更、リソース追加といったPJ全体影響のある判断	全ての管理領域を横断的に確認。 一歩先を見て、スケジュール遅延リスクや影響先を把握。 →足りない情報はリーダー層から回答を出させる。				影響先 PJメンバー全員 (全体の100%)
リーダー層 中程度の広さ	チーム内のユニット・サブチーム単位の状況確認 →直接影響のあるチーム間の連携	開発の品質面まで、チーム単位で確認。				チームメンバー (全体の20%)
メンバー層 狭く深く	個々のタスク進捗やチームメンバーへの影響確認					個々人の連携 (全体の3%)

開発と維持の並走

維持は即応、開発は計画重視—この優先度の違いが、予兆の共有を難しくする。
リソースの取り合いも重なり、予兆が見えていても対応が遅れてしまう。

リソースの取り合い

- ・リソースの取り合いにより、予兆が「一時的な人手不足」として処理される
→ 本質的な問題として認識されない
- ・リソース調整に追われ、予兆の検知・共有が後回しになる
→ 仮説構築や検証のサイクルが回らない

開発と維持保守が同時進行すると、
人員や時間が奪い合いになってしまうね。
各領域でトラブルも発生してるよ。



優先度の違い

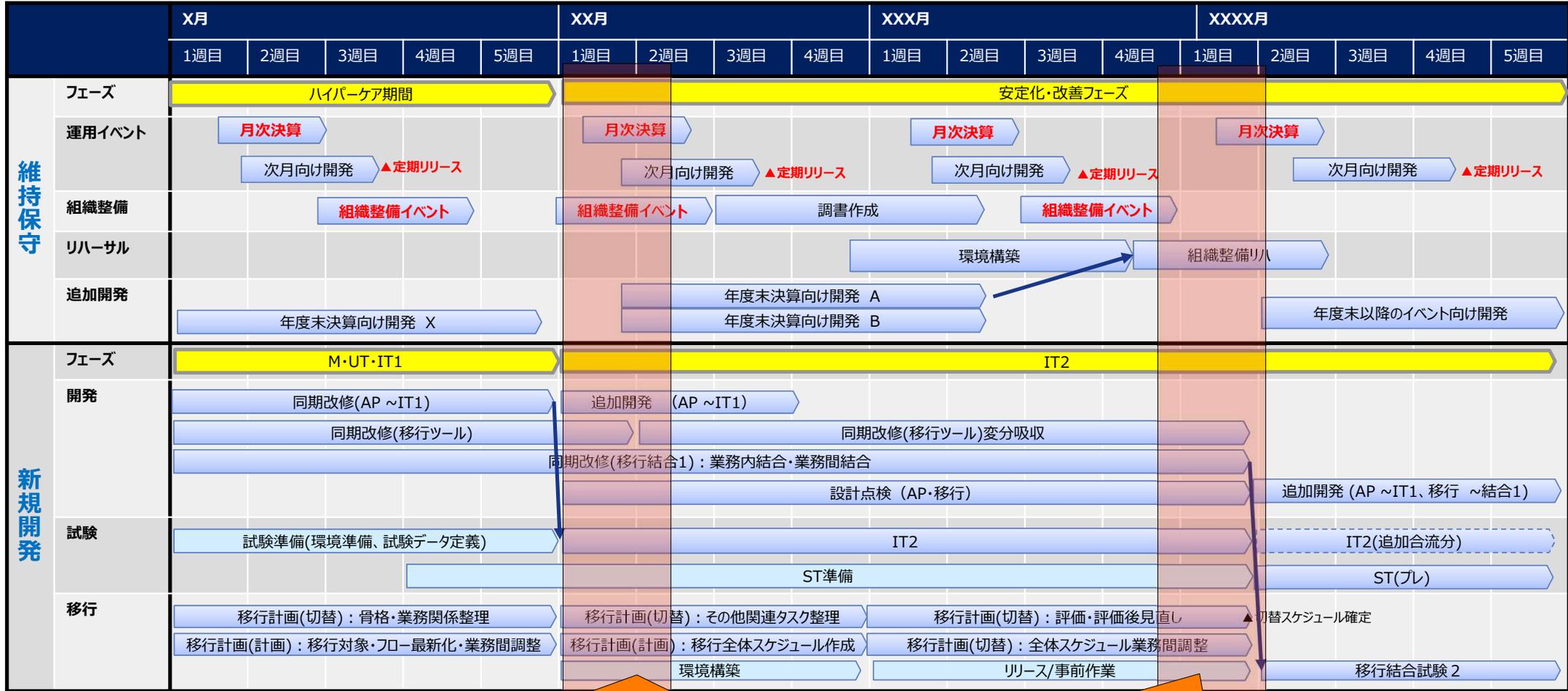
- ・維持業務：障害対応や運用イベントなど、即時対応が求められる短期的な視点
- ・開発業務：設計・試験・移行など、中期的な計画に基づいて進行する
長期的な視点。
→この時間軸の違いが、兆候の「捉え方」や「共有のタイミング」にズレを生む。



参考：開発並走に伴う課題

開発と維持のイベントが重なるタイミングが多く、リソース調整が困難となる。

※赤字：トラブル発生時に会社機能に影響有り。



開発のフェーズの境目が本体維持のイベントと重複しており、開発のフェーズ開始直後にリソースが確保できず。

開発のフェーズ完了時期も本体維持の重要イベントと重複。

1. プロジェクト運営の課題 まとめ

立場や業務の違いで、見たい情報も優先したいことも変わる。
予兆を捉えるには、視点に合わせた情報設計が必要になる。

視座毎のスコープの違い

プロジェクトを
全体俯瞰したい。



トップ層

チーム単位の状況を
把握したい。



リーダー層

個々のタスクの
全量・進捗を見たい。



メンバー層

開発・維持の優先度の違い

オンスケで進めたい。
中期的目線で、タスクの優先
度を見極めたい。



開発側

トラブルを即時対応したい。
短期的目線で、致命的トラブ
ルはとにかく最優先。



維持保守側

各層の方々が意思決定のために必要としている
情報をそれぞれ用意する必要がある。

開発・維持保守のタスクを並べた上で、
その都度優先度を見極められるようにしたい。

全体統括
PMO



1. プログラムマネジメントの難しさ

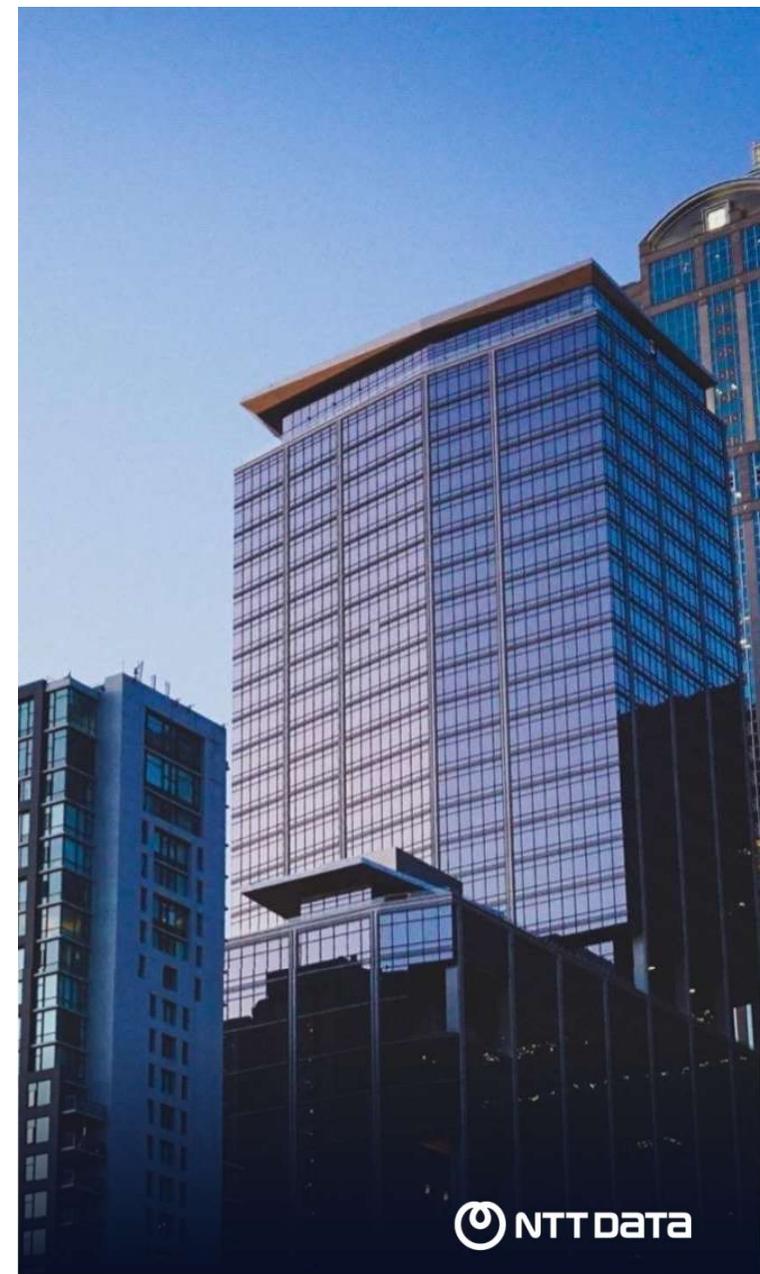
2. 当プロジェクトの課題と解決策

3. 実際の改善効果

4. リーダー層の声と妥当性の検証

5. 振り返りと今後の展望

当プロジェクトの特徴・課題と、
それらをどう対処したか報告します。



開発システム概要

当システム開発は、ServiceNowを用いた開発において、世界最大規模として紹介されている。

ServiceNow 公式HPでのご紹介

<https://www.servicenow.com/jp/customers/nttdatagroup.html>

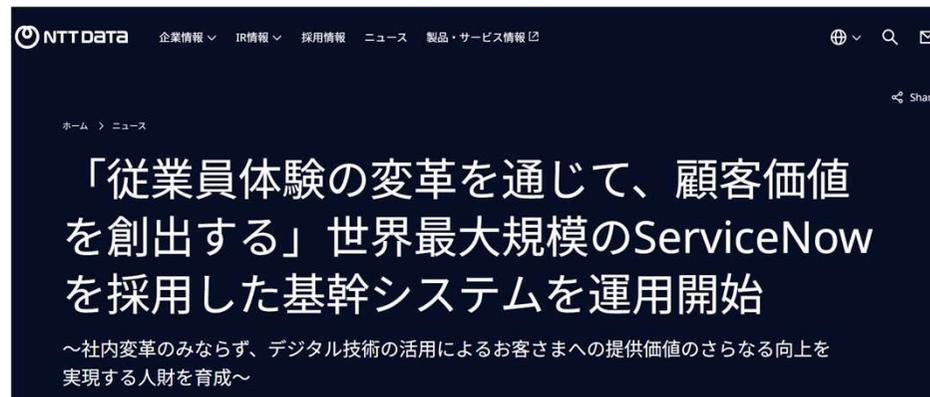


国内大手のシステムインテグレータ（SIer）として、公共・金融・法人の各分野で数多くのシステム開発を手掛けるNTTデータグループ。急速なデジタル化の進展を受け、自社システムにもデジタル技術を適用し、業務改革を目指す全社プロジェクト「Project GAIA」を展開しました。中でも注目されるのが、ServiceNowによる開発では世界最大規模※となる巨大な基幹システム、「GAIA.fin」（ガイア フィン）の構築です。

会社全体の決算はもちろん、各プロジェクトのマネジメントや意思決定のためにも用いられる非常に重要なシステムのため、これまではスクラッチで開発してきました。それをなぜ、NTTデータグループはSaaSであるServiceNowで構築することにしたのでしょうか？

NTTデータグループのニュース

<https://www.nttdata.com/global/ja/news/topics/2024/042601>



2024年4月26日

株式会社NTTデータグループ

株式会社NTTデータグループ（以下：当社）は、Project GAIA（以下：当プロジェクト）^{注1}の業務プロセス変革の中心として位置づけられた、GAIA.fin^{注2}において、ServiceNowを採用した基幹システム（以下：当システム）を運用開始します。

ServiceNow社が提供するローコードプラットフォーム「App Engine」を活用し、社内業務システムを統合した一元的なフロントエンドを構築し、効率的な業務プロセスを実現しました。当システムは、ServiceNowを活用したシステムにおいて開発規模および実行環境として世界最大規模です。

2024年4月初旬に一部サービスを先行して運用開始し、5月初旬に全面的な運用開始を予定しています。今後、国内グループ会社にもサービスを展開する予定です。

本プロジェクトの特徴

約600名の大規模プロジェクトで、維持/開発のリソース配分を各チームのリーダー層が判断している。
 → その中で、立場や優先度の違いにより、マネジメントは一層難しくなっている。

- 一般的なプロジェクト課題
 - ・トップ/リーダー/メンバー層の視座の違い
 - ・新規開発と維持保守の優先度の違い

- 本プロジェクト特有
 - ・参画メンバー約600名の大規模プロジェクト
 - ・新規開発と維持保守のリソースを対応チーム毎に配分。
 - ・対応依頼・指示がトップ層のみでなく、各統括チームから挙がってくる。

統括側

プロジェクトを
全体俯瞰したい。



トップ層

トップ層だけでなく、
各統括チームから指示・依頼が
発信される。

試験を
見ます。



各統括チーム

システム移行を
見ます。

業務側

チーム単位の状況を把握したい。
開発と維持保守のリソース配分したい。

もちろん自チームの
進捗管理もやらなきゃ。

リーダー層

Aチーム

変更管理
品質管理
進捗管理
スケジュール管理
故障管理
課題管理

開発担当

維持担当

リーダー層

Bチーム

変更管理
品質管理
進捗管理
スケジュール管理
故障管理
課題管理

開発担当

維持担当

リーダー層

Cチーム

変更管理
品質管理
進捗管理
スケジュール管理
故障管理
課題管理

開発担当

維持担当

リーダー層

Dチーム

変更管理
品質管理
進捗管理
スケジュール管理
故障管理
課題管理

開発担当

維持担当

リーダー層

E~XXチーム

変更管理
品質管理
進捗管理
スケジュール管理
故障管理
課題管理

メンバー層

自分のタスクで
精いっぱい。

各統括チームからの指示が
多くて、優先順位が
わからない。

自チームの管理で精いっぱい。
他チームの状況が見えない。

I/F
データ

シナリオ

シナリオ

I/F
データ

局所トラブルが全体計画に波及する構造

大規模プロジェクト故に、特定チームのトラブルが他チームに波及。
 ⇒ “釘一本で馬が倒れる” ようにトラブルが連鎖し、最終的にはプロジェクト全体計画にズレが生じる。
 こういった小さなトラブルの予兆を検知し、リスク想定→対策に繋げたい。

■ Aチーム (トラブル多発)
 ・進捗遅延が多発し、計画の見直しが必要。
 ・リソース不足が常態化。
 ・実行管理ができておらず、マネジメントに懸念。

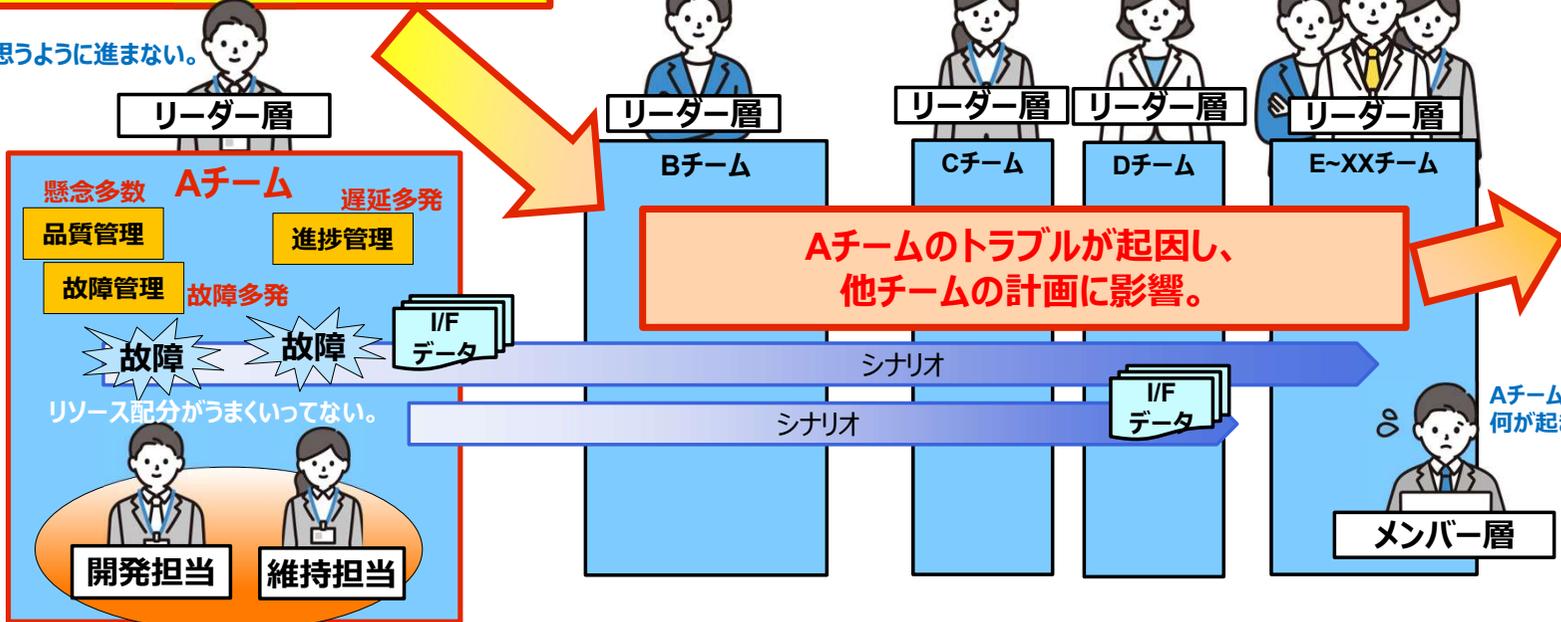
このタスクはAチームの完了待ちだな。

AチームやBチームのタスクが終わらないと、私たちのタスクも進まない。

Aで何が起きてる..?
 Bで..? Cで..? 他チームの状況が見えない。
 我々の対応が進まない。何があった?
 上からの指示待ち? 計画が...

プロジェクト全体の計画を見直さなければ。

思うように進まない。



Aチームのトラブルが、最終的に全体計画に影響

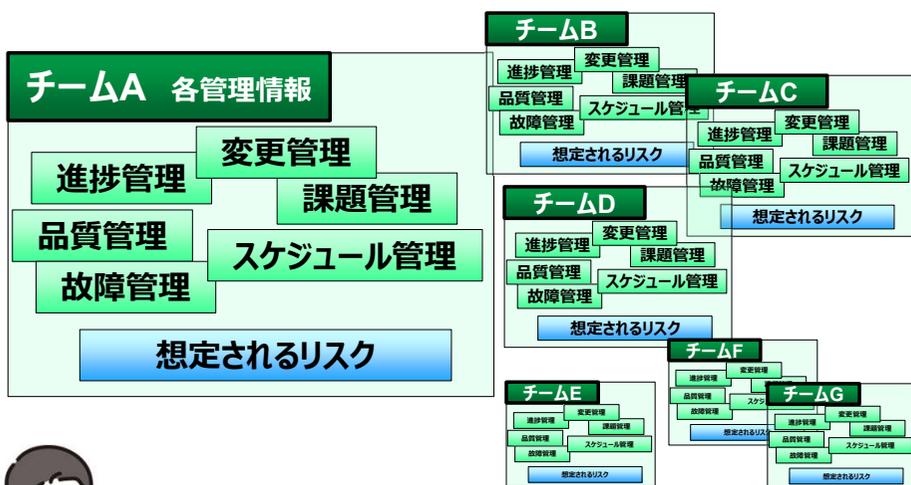


予兆検知の実現に向けた情報集約・可視化

トラブルの予兆を検知するためには、複数の管理領域の情報を統合的に見てリスクを想定する必要がある。

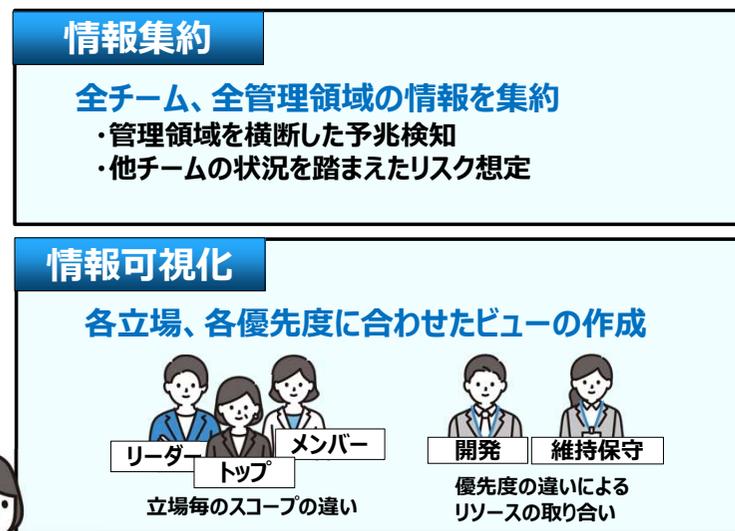
- ➔ 分散してしまっている情報を集約し、各視点に合わせたビューを作成することで、予兆検知に必要な材料を揃える。
予兆検知実現のためのキー情報となる管理ツール「**統合管理ビュー**」を作成。

AsIs 情報が分散していて予兆検知が困難



トラブルの種となる情報がチーム毎・管理領域毎に分散しており、これら情報を横断して予兆検知に繋げることが難しい状態です。

ToBe 情報集約と可視化によって予兆検知を実現



当プロジェクトでは、この情報集約・可視化した管理ツールを「**統合管理ビュー**」と名付けました。

予兆検知の精度向上に向けて試行錯誤を行い、情報の集約方法・可視化方法をブラッシュアップ。

予兆検知に繋がる情報集約の試行錯誤

複数の管理領域から情報を可視化し、仮説を立てプロジェクト推進を行っていく。
⇒ 予兆検知の精度向上のために試行錯誤を繰り返し、情報集約を進める。

Plan : 情報把握・分析

それぞれの管理領域の情報を集約し、
状況の把握・分析を実施する。

情報のキーとして、各視座・管理領域の情報を網羅的に集計・可視化
⇒ 管理ツール「**統合管理ビュー**」を作成。
PDCAを回すことで、この情報が洗練される。

Do : 予兆検知

管理領域を横断的に見て分析。

※前ページで提示した、各チーム・各管理領域を横並びにした、
横断的な見方による分析

情報を横断してみている、
重要なプロセスになります。



Act : 予兆検知の精度向上

予兆の仮説精度向上へ向け、改善検討。

- ・新たに見えてきた問題点や、可視化すべき情報を増やす。
※例. 故障改修の長期化が常態化したため、サイクルスピードを可視化。
- ・各管理領域から判断・検討する仮説の幅を増やす。
※例. 遅延状況の推移を可視化し、今後発生しうるリスクを評価。

⇒ プロジェクト推進を行う上で、可視化が必要な情報が増えていく。



仮説と分析のために必要な情報が取捨選択され、
予兆の仮説精度が向上していきます。

Check : 仮説検証

仮説が正しいか、現場の意見・状況と照らし合わせる。

- ・例. 立てた仮説を基にプロジェクト推進を実施。
⇒ 進捗会議の場で実際に課題提起を行う。

Doでの予兆検知の精度・妥当性を確認。

予兆検知に繋がる可視化方法の検討

集約した情報をもとに各視点で予兆検知を実現させるため、ヒアリングを通して用途・要望を確認。

➡ 予兆検知を実現させるための網羅的な情報共有が可能となる管理ツールを作成。

各視座の用途

上位層に何報告すればいいか知りたい。
自チームの状況を
知りたい。
自分のタスクはどれ？

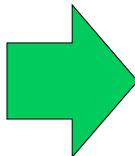
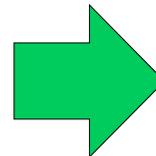
ビュー作成 効率化

ビューの作成に終日かかるから、手作業は無理。
マクロ・AIを使って効率化しよう。

ヒアリング（トップ層）

全体概況は文字ではなく、
○▲×でヒートマップで見たいです。 承知いたしました！

上記試行錯誤を通して、「**統合管理ビュー**」上で各視点に合わせたビューを作成

①. **PJ全体進捗概況（トップ層向け）**
どの管理領域で直近対応が必要なエスコレ事項があるか、○▲×のヒートマップで直感的に確認。

②. **Summary（リーダー層向け）**
各チームが該当週でどれだけエスコレ事項があり、各チームメンバーがどこを詳細に見なければならぬか確認する。

③. **各観点毎のビュー（メンバー層向け）**
エスコレ対象の具体的なタスクを記載。

補足：統合管理ビューの内容

トップ層向けのビュー

管理領域	品質	進捗	課題	品質	進捗	課題	品質	進捗	課題	品質	進捗	課題	品質	進捗	課題
品質管理	▲-●	▲-●	○	▲	○	○	○	○	○	○	○	○	○	○	○
進捗管理	○-▲	▲	○	○	○	○	○	○	○	○	○	○	○	○	○
課題管理	▲	X→▲	▲	X	○	○	○	○	○	○	○	○	○	○	○
変更管理	○	○	○	▲-●	▲-●	○	○	○	○	○	○	○	○	○	○
故障管理	○	○	○	▲	▲	○	○	○	○	○	○	○	○	○	○
スケジュール管理	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
品質管理	▲	▲	○	▲	▲	○	○	○	○	○	○	○	○	○	○
進捗管理	○-▲	○	○	○	○	○	○	○	○	○	○	○	○	○	○
課題管理	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
変更管理	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
故障管理	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
スケジュール管理	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

■ 直近で注視が必要な領域を直感的に把握

- 管理領域別のエスカーション状況
- ○▲×のヒートマップ表示
- 直近対応が必要な領域の可視化
- チーム別の注視ポイント（簡易）



今週、試験進捗が▲の理由は？

試験の項目漏れが見つかりました。



情報の
粒度

●、▲、×の基準は評価項目毎に定量的に設定。
評価項目・評価基準はPDCAを回しながら更新しています。

リーダー層向けのビュー

品質	進捗	課題	品質	進捗	課題
▲	○	○	○	○	○

メンバー層向けのビュー

品質	進捗	課題	品質	進捗	課題
○	○	○	○	○	○

■ 自チームおよび他チームの状況を把握

- 該当週のエスカーション件数
- チーム別の注視領域（進捗・品質・課題など）
- 各メンバーが見るべき観点の整理
- 他チームとの比較表示（相対評価）



■ 各メンバーが担当しているタスクの詳細情報

- エスカーション対象の具体的タスク一覧
- 試験進捗、故障対応、課題対応などの詳細
- 対応計画、滞留期間、品質懸念など



各管理簿の情報を一つのExcelファイルに集約
→シート毎に各立場に向けたビューを作成。

管理領域	管理簿・帳票名
進捗管理	・各業務別一覧（0000進捗表） ・品質・進捗別進捗表 ・品質管理
品質管理	・品質管理表（品質管理表） ・品質管理表（品質管理表） ・品質管理表（品質管理表）
課題管理	・品質管理表（品質管理表） ・品質管理表（品質管理表） ・品質管理表（品質管理表）
変更管理	・品質管理表（品質管理表） ・品質管理表（品質管理表） ・品質管理表（品質管理表）
故障管理	・品質管理表（品質管理表） ・品質管理表（品質管理表） ・品質管理表（品質管理表）
スケジュール管理	・品質管理表（品質管理表） ・品質管理表（品質管理表） ・品質管理表（品質管理表）

集約

Excelファイル

情報の
可視化



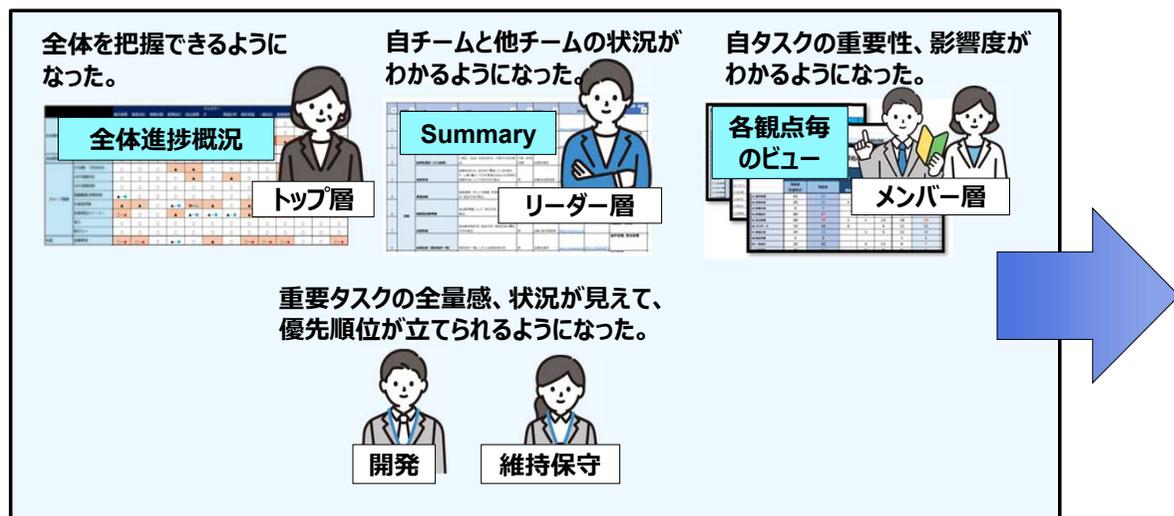
リーダー層向け

高

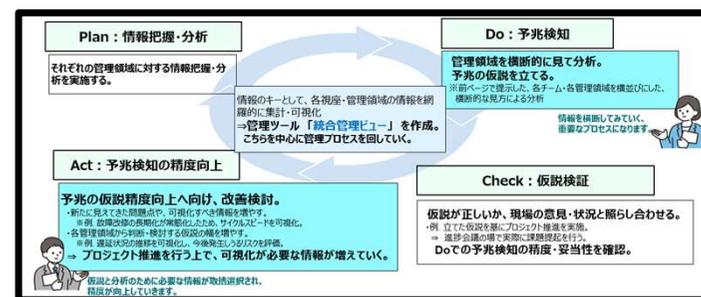
統合管理ビューを通したマネジメント改善の実現

各管理領域を複合的に分析することで、予兆検知に繋がる重要な情報を可視化。
 ⇒ 可視化した情報を基に予兆検知を行い、大規模プロジェクトのマネジメント改善へと繋げる。

予兆検知実現に向けた試行錯誤を行い、
 統合管理ビューを作成



予兆検知実現に向けた試行錯誤を通して、
 会議運営・品質・チームマネジメントを改善。

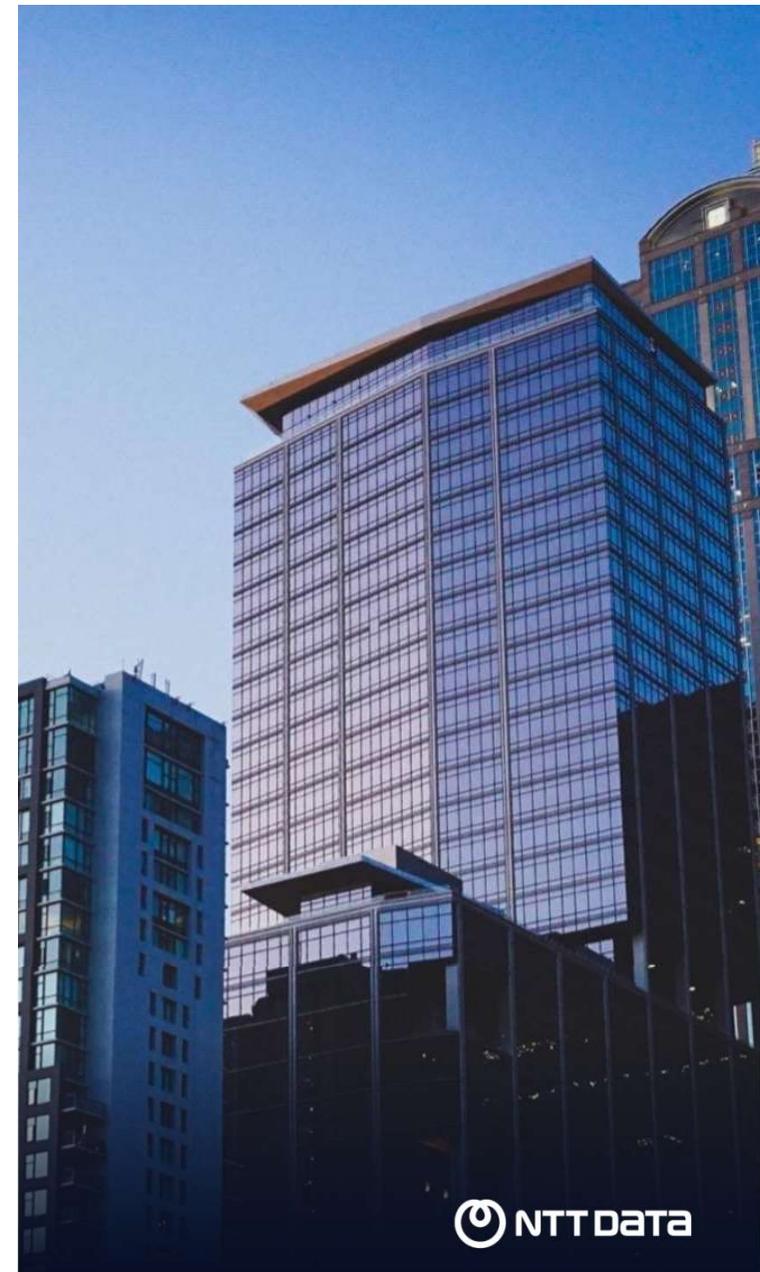


統合管理ビューを通した予兆検知の精度向上のため、
 継続してPDCAサイクルを回し、マネジメント改善へと繋げる。

- Plan : 情報が整理され、チームマネジメントの基本動作の質向上
- Do : 兆候の仮説・分析により、リスクの判断精度が向上
- Check : 会議が課題を拾う場から、議論する場へと変化
- Act : 仮説検証を踏まえ、マネジメントの仕組みと行動を改善・定着

1. プログラムマネジメントの難しさ
2. 当プロジェクトの課題と解決策
- 3. 実際の改善効果**
4. リーター層の声と妥当性の検証
5. 振り返りと今後の展望

管理プロセス改善により
どう成果を出したか、報告します。



Plan : 予兆管理の出発点

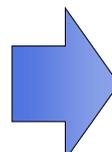
兆候は複数の管理領域に分散して現れるため、個別管理では“兆し”として認識されず、対応が後手に回るケースが多発。

➔ 管理領域を横断的に整理・統合することで、マネジメントの基本動作（把握・判断・共有）を徹底

管理を徹底したいけど、
管理簿が多くて情報が整理できない。



チームA リーダー



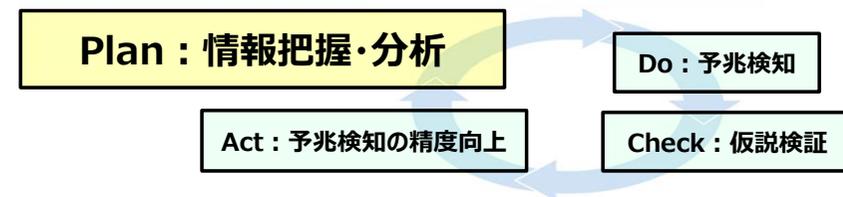
全体統括
PMO

進捗、品質でタスクの
滞留がありますね。



チームA リーダー

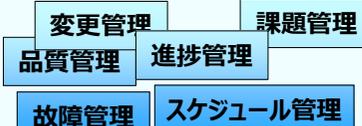
今週注視すべきところが
明確になりますね。



Do : 横断的分析による兆候抽出

管理領域を横断したことで新たに見える情報が増え、予兆の仮説・対応策の検討に繋がる。

管理領域を横断的に確認



新たに見える情報

Plan : 情報把握・分析

Do : 予兆検知

Act : 予兆検知の精度向上

Check : 仮説検証

これらの詳細状況を
会議で確認しよう。



例	管理領域	予兆	対策
① 進捗報告は順調だが、実績工数が異常に少ない	品質×進捗×工数	<ul style="list-style-type: none"> 品質レビュー未実施 テスト未着手の可能性 	レビュー漏れを検知し、試験実施前に是正
② 課題未登録で進捗停滞	課題×進捗×故障	<ul style="list-style-type: none"> 実行管理の不全 	チームマネジメントに懸念 →支援を実施
③ 進捗は順調だが、対応計画が後半集中	課題×スケジュール×品質	<ul style="list-style-type: none"> 品質問題の顕在化リスク 	計画のフィジビリティ確保 事前分散対応を促進
④ 故障改修が長期化しており、進捗報告は「対応中」のまま	故障×品質×生産性	<ul style="list-style-type: none"> 根本原因未解決 再発リスク 	サイクルスピードを可視化 →対応計画の見直し

Check : 仮説検証による会議運営の質向上

進捗会議の場で予兆の仮説を提示し、現場の状況と照合することで、兆候が実際の課題に繋がっているかを検証。

➔ 会議の場で課題を拾うのではなく、重要課題が顕在化前に議論ができる状態に。

Plan : 情報把握・分析

Do : 予兆検知

Act : 予兆検知の精度向上

Check : 仮説検証

予兆の仮説を会議の場で確認

例	管理領域	予兆	対策
① 進捗報告は順調だが、実績工数が異常に少ない	品質×進捗×工数	・品質レビュー未実施 ・テスト未着手の可能性	レビュー漏れを検知し、試験実施前に是正
② 課題未登録で進捗停滞	課題×進捗×故障	・実行管理の不全	チームマネジメントに懸念 →支援を実施
③ 進捗は順調だが、対応計画が後半集中	課題×スケジュール×品質	・品質問題の顕在化リスク	計画のフィジビリティ確保 事前分散対応を促進
④ 故障改修が長期化しており、進捗報告は「対応中」のまま	故障×品質×生産性	・根本原因未解決 再発リスク	サイクルスピードを可視化 →対応計画の見直し

冗長だった報告が絞られ、短く・濃い議論に。

- ・重要課題に絞った報告・議論となり、対策の検討や関係チームとの連携を行うことが可能となった。
- ・会議時間そのものが短く、濃くなった。

従来 : 3.0h (延長も多発) → 現状 : 2.0h (時間超過ほぼ無し)



リーダー層

会議で議論すべき重要課題に絞って報告

今まで思いつくまま会議で報告していたが、重要なトラブルの兆候が見えて、報告内容が絞られた。



トップ層

注視すべきポイントの明確化

各管理領域・チームを横断して見ているので、どこに問題があるか予測がつけやすい。



全体統括
PMO

会議の運営品質の向上

こちらから課題提起ができたことで、ただの司会進行役から、会議運営を主体的に握れるようになった

Act : チームマネジメントの抜本的改善

予兆検知をした上で会議で議論を行うことで、新たに見るべき情報や分析軸が追加される。

➔ 仮説検証のプロセスを回すことで“気づき”が増え、マネジメント行動の変革と定着に繋がる。

チーム毎のマネジメント課題

リソースが不足し、遅延が発生



課題：維持・開発のリソース調整

複数の管理領域で遅延発生。



課題：チーム全体の計画再調整

管理簿が更新されず、滞留多発。



課題：管理簿の最新化

Plan : 情報把握・分析

Do : 予兆検知

Act : 予兆検知の精度向上

Check : 仮説検証

予兆検知を実施したことで、特定のトラブル・課題の特定だけでなく、抜本的なマネジメント課題を可視化することが実現。

	チームA	チームB	チームC	チームD	チームE	チームF	チームG	チームH	チームI	チームJ	チームK	チームL	チームM
問合せ	▲	×	○	▲	▲→	○	▲	▲→	○	○	○	○	○
メンテナンス影響	○	▲	○	○→▲	○	○	○	○	○	○	○	○	○
経路変更	▲	×	○	×	×	▲	▲	▲→	×	▲	▲	○	○
運用	○	○	○	○→▲	○	○	○	○	○	○	○	○	○
対応計画	○	○	○	○	○	○	○	○	○	○	○	○	○
経路変更	○	○	○	○	○	○	○	○	○	○	○	○	○
問合せ	▲→	▲→	○	○→▲	○	○	○	○	○	○	○	○	○
生産性評価	▲	○	○	▲	▲	○	○	○	○	○	○	○	○
品質評価	○	○	○	○	○	○	○	○	○	○	○	○	○
課題	○	○	○	○	○	○	○	○	○	○	○	○	○
仮説検証	○	○	○	○	○	○	○	○	○	○	○	○	○



各管理領域の情報を横並びに可視化したことで、さまざまな兆候や課題に気づきました。

マネジメントとして意識しなければならない部分の気づきや、予兆検知の精度向上のためにプロジェクト全体で見ると必要な情報が増え、プロジェクト全体のマネジメント品質が向上していく。

予兆を捉えないとまた遅れが出るな。意識しよう。



リーダー層

進捗や品質だけでなく、開発の対応スピードも毎週可視化する必要があるな。



トップ層

3. 実際の改善効果 まとめ

PDCAサイクルを通じて、予兆検知を行う“仕組み”をマネジメントに定着させることができた。

➔ 大規模・多層構造の本プロジェクトにおいて、統合管理ビューを通してマネジメントプロセスの改善に成功。

一般的なマネジメント課題



統合管理ビューを通して予兆検知を実現。
一般的なマネジメント課題に応じ、大規模・多層構造プロジェクトにおいてマネジメントプロセスの構築を実現。

- ・管理領域を横断的に見ることで、兆候の発見精度が向上
- ・仮説検証を通じて、意思決定の質とスピードが改善
- ・行動変容が促され、マネジメントの対応力が向上
- ・予兆管理がプロジェクト運営の“当たり前”として定着

本プロジェクトの特徴



複数のチームが同時に開発を進め、維持業務も並行して走る—
そんな複雑な環境で、

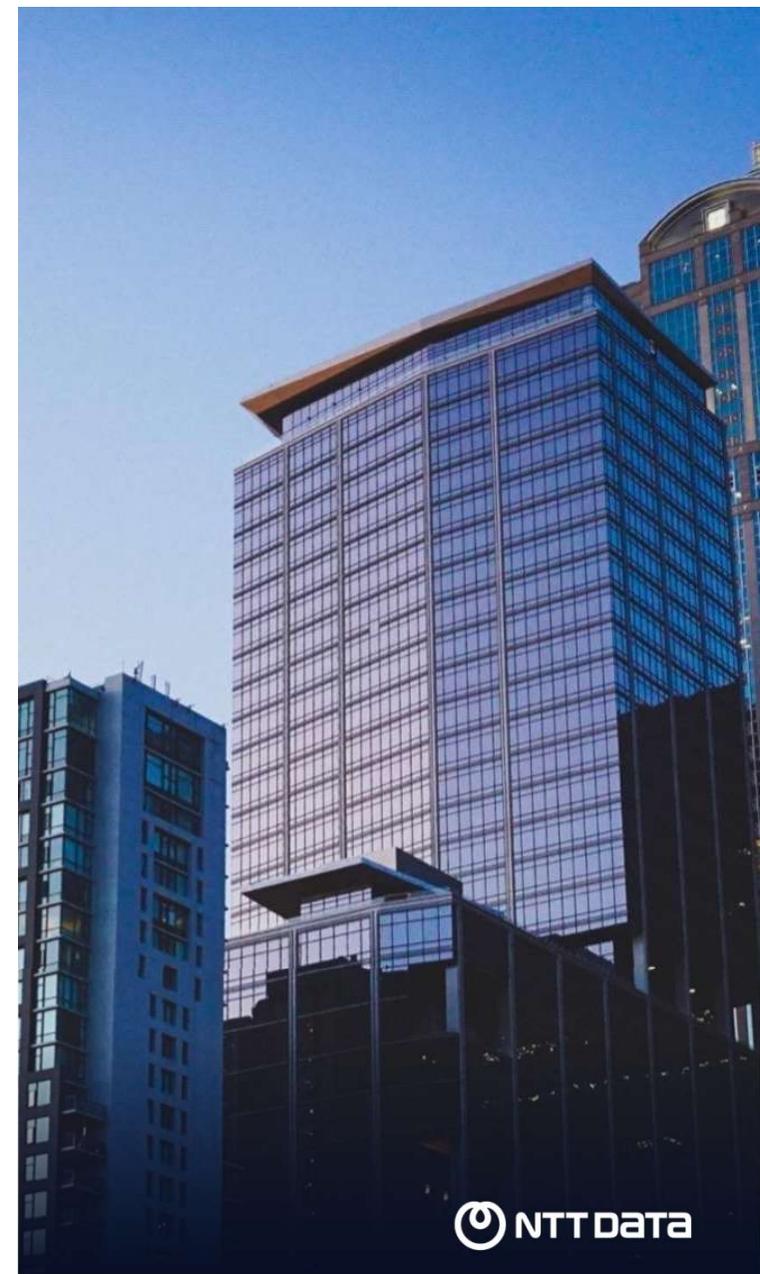
プロジェクト運営を行うことが可能となる管理プロセスを構築。

1. プログラムマネジメントの難しさ
2. 当プロジェクトの課題と解決策
3. 実際の改善効果

4. リーダー層の声と妥当性の検証

5. 振り返りと今後の展望

実際に運用されている方々から
ヒアリングを行いました。



各リーダー層の声

管理ツールの新規導入によるプロセス改善は、プロジェクト全体の“気づき・判断・行動”を支える基盤として、リーダー層からも高い評価を得ている。

○コミュニケーション面

- ・報告観点が明示されたことで、**資料作成の負荷が軽減され、報告の質が向上。**
- ・他チームとの課題突合がしやすくなり、**横断的な連携がスムーズに。**
- ・第三者視点での定期評価により、**自チームの立ち位置を客観的に把握**できた。

○マネジメント面

- ・チェック観点や重点監視項目が明確になり、**チーム内で目線を揃えた運営が可能に。**
- ・リーダー層の管理工数が削減され、**業務仕様・運用検討に集中できる環境が整った。**
- ・課題の抜け漏れが防止され、**対応の迅速化に繋がった。**

○予兆管理面

- ・定量化された情報により、**自チームでは気づけなかった懸念点を早期に察知。**
- ・毎週の進捗確認で、**他チームとの比較や傾向把握が可能になった。**

状況把握が
楽になった。

会議報告がスムーズ
になった。



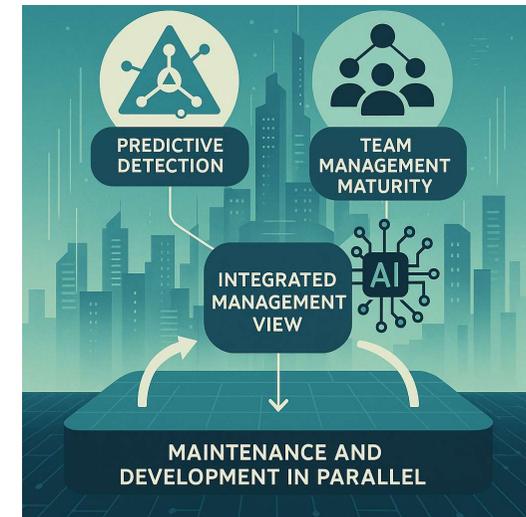
振り返りと今後の展望

■ 振り返り ～複雑な環境下での統合管理の試行錯誤～

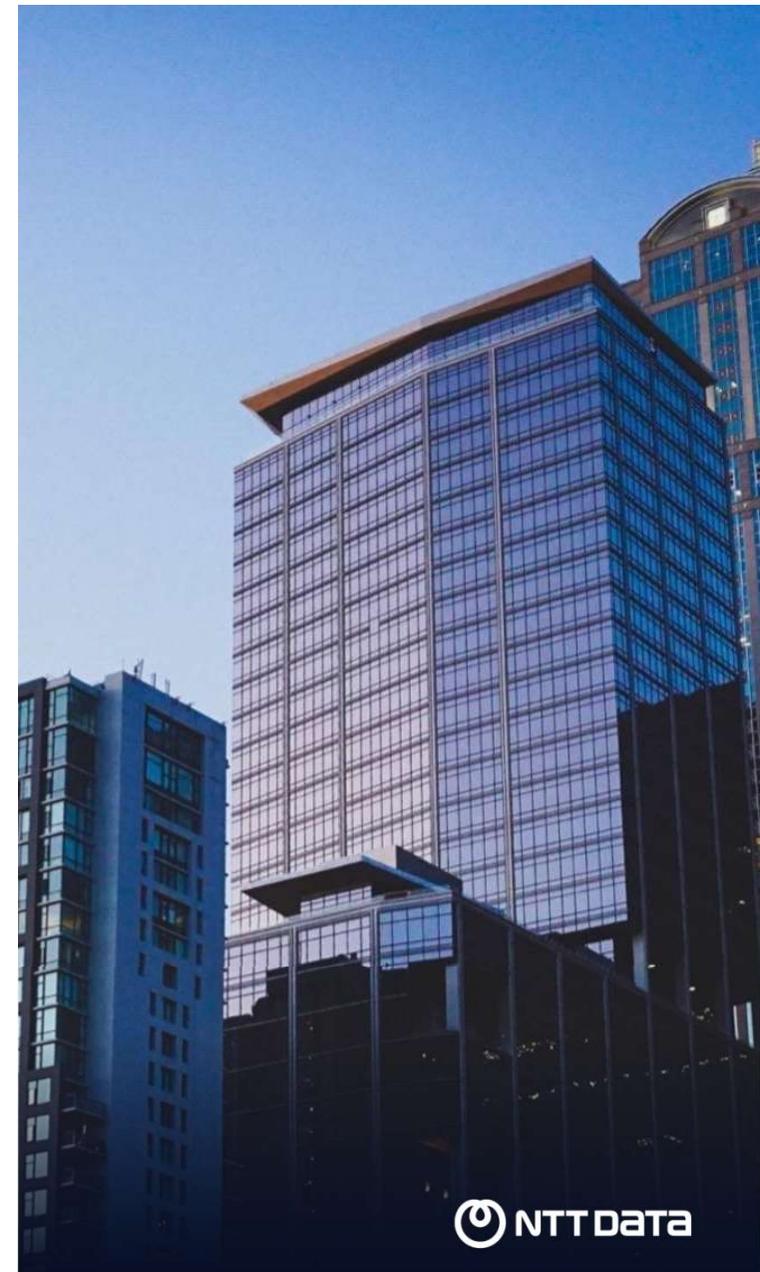
- ・**維持と開発の並走**というプロジェクト運営の難しさに対し、従来の管理手法だけでは限界があることを実感。
- ・**予兆検知の重要性**を再認識し、リーダー層・メンバー層それぞれの視点を意識した情報設計を実施。
- ・**統合管理ビューの進化**を通じて、プロジェクト全体の可視性と先読み力が向上。
- ・各チームの**マネジメント成熟度の違い**を受け入れつつ、共通の管理基盤を模索。

■ 今後の展望 ～より強靱なプロジェクト運営へ～

- ・**統合管理ビューのさらなる進化**
 - AIや自動化ツールの活用による予兆検知の高度化
 - チーム間の情報連携を強化し、全体最適を目指す
- ・**マネジメントスキルの平準化**
 - TL/GL層へのナレッジ共有や教育機会の拡充
 - 成熟度に応じた支援モデルの構築
- ・**当マネジメントプロセスの横展開**
 - 他プロジェクトへの適用可能性を検討
 - 並走型開発におけるベストプラクティスの確立



ご清聴ありがとうございました。



The image features a low-angle, wide shot of a modern city skyline under a clear blue sky. Two prominent, tall skyscrapers with white facades and dark window bands are the central focus. Other buildings of varying heights and architectural styles are visible in the background. In the foreground, there are some trees and a street with a few vehicles. Overlaid on the center of the image is the NTT Data logo, which consists of a white circular icon with a stylized 'D' inside, followed by the text 'NTT DATA' in a bold, white, sans-serif font.

NTT DATA