

『大規模開発管理』について

SI 事業部 技術 2 部

細 川 浩 治

1. はじめに

システム開発において小規模から大規模まで開発総工数により分類されるが、大きくなればプロジェクトの体制も大きくなり、リスクも比例して大きくなる。100人月オーバーのプロジェクトでリスクの読みを間違えると大きなダメージを受けることになる。システム開発のプロジェクト管理とはまさに問題と解決をいかにコントロールするかであり、規模が大きくなればなるほど問題点も多くなり、その対応如何によりリスクも上下する。プロジェクト管理手法により、このリスクを分散し軽減することが可能である。以下にその手法を要点のみ述べることにする。ただし、特別な管理手法を用いた訳でもなく、ごくあたり前の管理手法であり、小規模からのシステム開発でも問題なく適用できる。

2. 役割

プロジェクトを形成するメンバーは、以下のような体系となる。

基本的には、すべての役割をアサインこと。

開発規模とプロジェクト要員は、揃えること。

適材適所の要員をアサインすること。

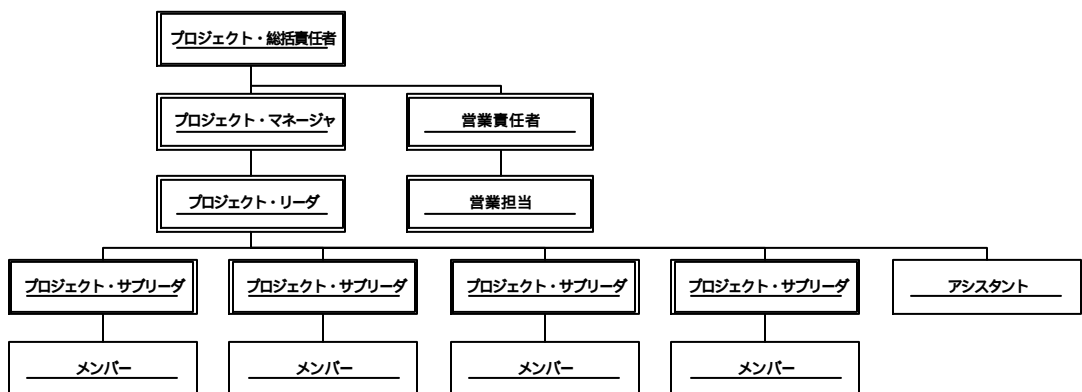
指揮命令系統を体系の通り定めること。

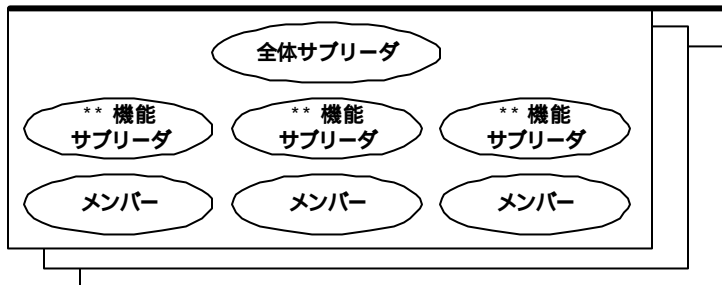
メンバーに対する責任者 (指示・管理) は、サブリーダーである。

サブリーダーに対する責任者 (指示・管理) は、プロジェクトリーダーである。

マネージャは、プロジェクト全体の責任者である。

プロジェクトを形成する場合、役割分担を明確にすることが必要である。誰がリーダーで誰がサブリーダーで...、作業範囲と権限・責任を明確にすることで





設計チーム / 製作チーム / 試験チーム

工期が比較的長いもの、大規模なものは、設 / 製造 / 試験の大きく3つのチームに分類し、それぞれにサブリーダーを配置させるとよい。また、それぞれのフェーズの中は、機能単位に細分化し各人の役割を明確化させる。

プロジェクトは、やはりリーダー・サブリーダーがいろんな意味で大きな影響を与える。一般的に、リーダーがしっかりしていればなんとかなるものだが、大規模になればなるほど要員数も増しサブリーダーの資質も問われる。ただし、理想のアサインはできないのが実情であるため、チームとして役割が重要となる。

3. 各フェーズでのチーム編成について

大規模システムでは、是非とも死守してほしいのが、設計・製作・試験のメンバーにある程度は、重複しないメンバー(専任)をアサインすることである。設計したものがそのまま製造するのが効率的でいいのではないかとの意見もあるが、設計チームは製造チームへ設計書を介して作業を引き継ぎ、ここで設計品質が確保できる。いい加減な設計だと製造チームからクレームが挙がることになる。さらにもう一つの効果としてチーム間におのずとコミュニケーションが発生し相乗効果

を産むことになる。

システムを同じ視点で見ることも必要だが、設計の立場から、製作の立場から、試験の立場から、それぞれの立場から見ることも重要であると思う。

4. コミュニケーション

プロジェクトとして一番大事なものがコミュニケーションである。コミュニケーションがないプロジェクトはその時点で大きな障害を抱えており、コミュニケーションに問題があると必ずトラブルが発生する。プロジェクトの中でいろいろな役割(立場)のメンバーが存在することになるが、必ず毎日ミーティングを行うことが必要である。これは、全員で集まると非常に人数が多いことになるのと、ミーティングのテーマが設定し難い面があるので、あるグループ単位に実施すればよい。

また、あるサイクルでできる限り多くのメンバーとコミュニケーションを図れるようにミーティングを計画することが必要である。

5. 情報交換と整理

ユーザとの情報のやり取りおよびプロジェクト

内での情報交換の情報量はプロジェクトが大きくなればなるほど、情報量も比例して大きくなる。この情報整理がプロジェクトの進捗および品質に大きな影響を与えることになる。基本的に、すべての情報を電子データとして管理することを前提とする。

ユーザとの情報交換

ユーザとの情報交換の媒体としては、共通フォーマットの用紙を使用する。

Ex)

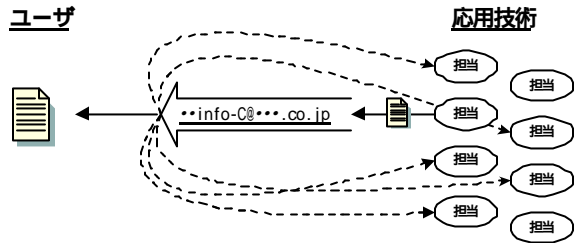
・連絡票；仕様等の連絡 / 確認 / 質問等...

・障害票；障害連絡

・リリースノート；リリース時のリリース情報
etc...

情報により ML (Mailing list) を作成し、この ML を介してやり取りする。

情報により、関係する担当者へ mail が送信されるようにする。



特にユーザとやり取りする情報に関しては、透明性を重視するとともに、特定個人の情報と取得ないように関係担当者へ送信するようにする。

プロジェクト内の情報交換

基本的には、コミュニケーションを重視するため、打合せ形式による情報交換を前提とするが、些細な確認事項および事前周知事項 / 連絡等、すべて ML にてやり取りを行なう。プロジェクトに係わる情報は、原則、些細なことでもすべて関係者に連絡が伝わるようにする。情報の共有化と ML にて流れる情報により、発信者の状況を確認することにも役立つ。

注意してほしいのは、ML にてやり取りすることが基本ではなく、情報交換は打ち合わせ形式で行うことが前提であり、ML は極端に言えば、その打ち合わせ結果をメンバーに周知させるために使用することである。

Ex)

各チーム内 ML

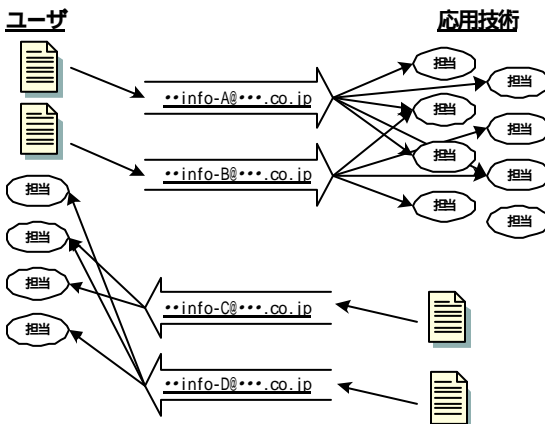
基本設計 ML

詳細設計 ML

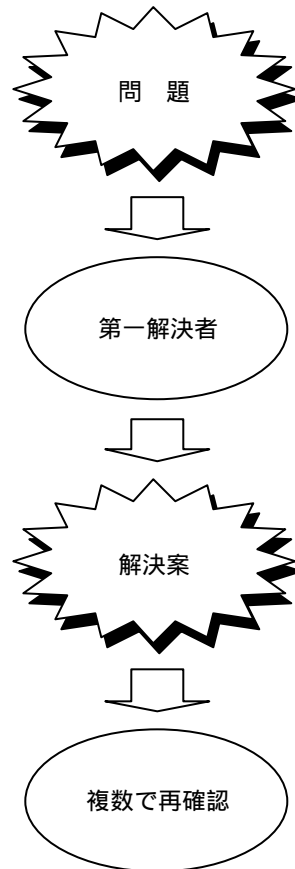
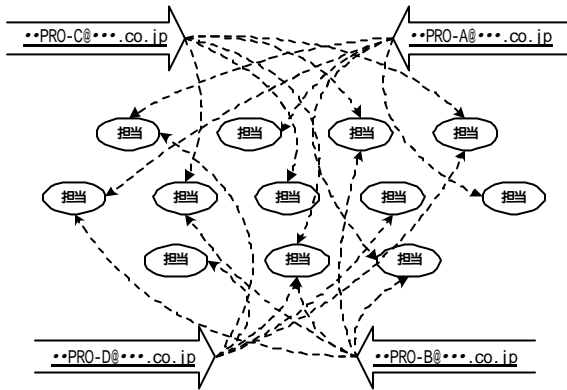
障害 ML

全メンバー ML

etc...



社内では、ユーザに送信した mail は、必ず関係担当者にも送信されるようにする。



6. 問題点の解決手法

プロジェクトの中では、ユーザ間を含め日々多くの問題が発生する。問題はすべて解決することが前提となり、問題の置き去り・間違った解決は、プロジェクトに何らかの影響を与え、小さなことでも積み重なれば大きなダメージを受けることにもつながる。

解決手法としては、非常に単純であるが、問題に対する解決者（特定メンバーが選定できない場合は、複数関係者を選定する。）を選定し、解決予定期日を定め解決にあたる。解決した内容は、関係者により再度、解決内容を確認する。

要は、一人の判断では解決とみなさない。必ず複数のメンバーにより最終判断を下す。

問題の解決時の対応として常に問題のもつ影響範囲を予測する必要がある。特に工期に影響する問題であれば、最終的な工程までシミュレーションし影響を検討する必要がある。一般的に工程に影響が出る問題は、最後のフェーズで調整せざるを得ない状況になりがちであるが、いつかは調整不可能な状況になり、多少なりと後工程まで尾を引きそうな問題は、要注意である。

7. 意思決定

いろいろな場面で判断（決断）することがあるが、判断を下すことのできるものは仮に判断が間違った場合に、責任（対処）が取れることが条件となる。この条件を守れない可能性があれば、当然ながら上位のメンバーに判断を仰ぐことが必要である。基本的には、ある程度、影響度のある判断については、複数関係者による判断を原則とすればよい。

また、判断のレスポンスは重要であり、ひとつの判断の遅れが及ぼす影響は常に把握（予測）し、できる限り早めに対処することが理想である。

8. 規約 (ルール)

プロジェクトが大きくなれば、要員もそれだけ多くなる。このメンバーをコントロールするには、リーダーが声を張り上げて制御しても限度がある。

前述の情報交換で ML の活用もそうであるが、作業手順を規約化することも重要である。組織もそうであるように、あるルールに基づいて活動している。プロジェクトも同じようにルールを作り、これに従うべきである。ただし、この規約は運用して問題があれば、臨機応変に変更し、常に状況に応じたものに変更することが必要である。

Ex)

連絡票運用規約

障害管理規約

工程会議規約

開発環境規約

・コーディング規約

・ライブラリ管理規約

etc...

9. おわりに

プロジェクト管理としてリスク分散・軽減が大きな課題であるが、単純なことでもなかなか実行することが難しいのが現実である。マネージャ (リーダー) としてプロジェクトを遂行する場合、やはり経験が重要である。

経験の中でもいろいろな局面においての判断能力・予測能力を身に付けることが重要である。プロジェクトの構成要員は“人”であり、プロジェクトを動かすのも“人”である。デジタル的な対応も必要となるが、経験を重ねてほしい部分は、や

はりアナログ的な対応であり、言葉での表現は非常に難しいが、あえて言うなら“勘”ということになる。

マネージャ (リーダー) は、日々この“勘”という微妙な感覚を発達させ精度を高めることにも努力してほしい。

最後に非常にあいまいな表現となってしまったが、システム開発の中でプロジェクト管理は、管理する内容・手法等、明確に定義することは可能であるが、いざ実行となると、なかなか思うようにいかないものである。しかし、システム開発を行うということはプロジェクトに存在していることであり、いつでも管理手法について見て習得できる環境にある。メンバーはサブリーダー、をサブリーダーはリーダーの管理手法について自分なりのシミュレーションをしたり、打ち合わせの場等で積極的に自分の意見を出すようにしてもらいたい。