

## 2.19 システム開発現場のコミュニケーションとモチベーション向上に関する教訓 (G19)

教訓  
G19

## みんなで唱和！障害減らす教訓共有

2

ガバナンス／マネジメント領域の教訓

## 問題

A社では、A社の開発部門が構築した制御装置を監視制御する物流拠点内の配送自動化システムを運用している。A社の開発部門は、同システムを、長期にわたり物流ニーズの変化に合わせた保守開発を行っている。

ある日の早朝にこのシステムの制御装置の1台にシステム障害が発生した(図2.19-1)。

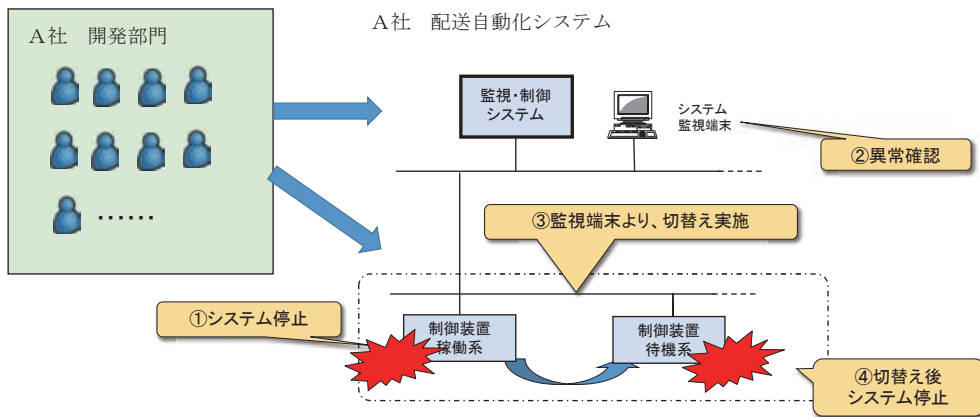


図 2.19-1 障害状況

制御装置の稼働系が障害になり(図2.19-1①)、システム監視端末が、異常を検知した(図2.19-1②)。システム運用者は、端末の表示メッセージからハードウェアの問題と思い、すぐに、制御装置を待機系に切り替えた(図2.19-1③)。しかし、切替え後も障害となった(図2.19-1④)。

運用管理者は、すぐに原因が分からなかったが、システムを一旦リセットすることにした。そこで、システム全体を再起動したところ、稼働系の制御装置は正常に戻った。

## 原因

直接原因は、制御装置プログラムが持っている制限値オーバによる制御装置の停止であった。そのため、制御装置を待機系に切り替えても、待機系の制御装置も制限値オーバで停止してしまった。

根本原因は、今回の障害となった制御装置が持つ制限値は、システム構築当初から存在していた

ものだったが、このような制限値があることを知っていたのは、この個所の設計、製造を受け持った一部のメンバだけであった。そのため、この制限値が開発チーム全体で共有されることがなかったため、障害復旧が遅れてしまった。

## 対策

今回のような「担当者しか知らない」仕様ががあったために起きた障害を2度と起こさないために、A社の開発チームは、「どこに問題があったのか」、「対策は何か」等、何回も議論を重ね、アイデアを出し合って、原因、対策をまとめた。その過程で課題がいくつか浮かび上がった(図 2.19-2)。

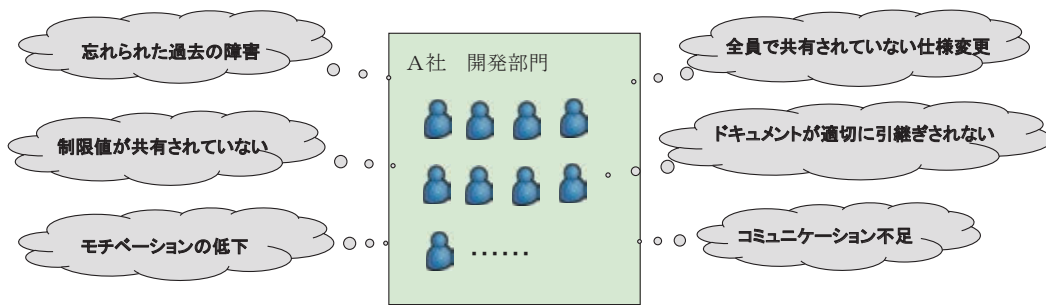


図 2.19-2 浮かび上がった課題

これらの課題にどのように対応するかを検討したときに、教訓として残し、それを常にメンバが意識できるようにしておくことが、新しいメンバが参加しても風化させないことにもなると考えた。そこで、開発チームの代表メンバで「教訓」を作成し、開発チーム内全員が集まったときに、その「教訓」をみんなで声を出して読み合うこととした。

具体的には、以下の「教訓」をまとめ、毎日の朝会(開発チーム全員への伝達事項等の共有を図るミーティング)の開始前に、全員で唱和している。

なお、「教訓」は、大きな障害が起こるごとに、チーム内で議論し合い、その度に追加・更新していった。

### みんなが困る! 知らない仕変(仕様変更)

システムやプログラムの制御において、テーブルのサイズ等には上限を定めることが一般的である。そして、この制限値は、システム間やプログラム間でも相互に影響をもたらすことがある。例えば、あるプログラムの制限値を拡張した場合、そのプログラムと連携する次のプログラムにも影響することが多々ある。したがって、この制限値がどのように決められているのか、チームメンバが知っているのか、などの点を管理することがシステム障害を減らすポイントである。そこで、この開発チームでは、「みんな

なが困る! 知らない仕変」と各人が肝に銘じておくことに決めた。これによって、今後制限値変更があった場合、その変更内容を全員が共有できることになった。

### みんなが実施! 着実なハウレンソウ

システム障害対策においては、チーム全員の連係が重要である。特に多くの開発メンバで仕事をしている場合は、誰の担当個所の障害かを判断することや、各人の担当パートをつなぎ合わせてどこが障害かを探すことなども必要となる。そこで、全員の間で情報伝達が、早く正確にできることを目標に、着実な、「報告、連絡、相談」を行おうとの決意を込め、このようなメッセージとした。

### みんなで共有! 常に最新、設計書

仕様変更などが起きた場合の情報共有のポイントは、「何をもとに共有するのか」になる。そこで、開発チームの共有すべきものを設計書とした。一旦、設計書を情報共有の基盤としたことによって、システム更改に合わせて設計書の改版を行うことは、開発チーム全員にとって重要な作業となった。このような教訓メッセージを定めることにより、ソフトウェアの変更は行ったが、設計書の改版は後回しにしてしまったり、改版を忘れてしまったり、などともなう情報の不整合をなくすことに努力した。

### みんなで確認! その疑問

システム障害が起こる前には、なんらかの予兆が存在していることが多い。そのひとつが、開発チームひとりひとりの「この点はどうなのだろうか?」、などの疑問から生まれる「気づき」ではないかと考え、「みんなで確認! その疑問」とした教訓メッセージを定めた。この教訓メッセージを定めることにより、チーム内で確認し合うことに遠慮がなくなり、コミュニケーションが積極的に取られることを目指した。

### みんなで目指せ! 変化に敏感、気づく人

上記の「みんなで確認! その疑問」を実践するためには、ひとりひとりの「気づき」が発揮されないと実践できない。やはり、基本はチームひとりひとりのモチベーションの向上が欠かせない。そのためにも、ベテランメンバから新規参画メンバへの教育や、意見交換などの日ごろの活動が重要になってくる。モチベーション向上の意味でも、この教訓メッセージの全員での唱和は、重要な活動のひとつとして、定着している。

## 効果

このような活動を通して、開発チーム内では、システム障害が起きて対応が一段落するごとに、反省とこれからどう対策を行うかといった、メンバひとりひとりのシステム障害への積極的な取り組みが生まれ、自然と、みんなで話し合う風土が醸成されることになった。

朝会など、チームメンバが集まった場で、チーム全員で「教訓」を唱和することにより、チーム内のコミュニケーション向上と、各人のモチベーションの向上が図られる。

また、システム障害時の経験で学んだことで得た「教訓」をみんなで共有することで、その経験を風化させないことができる。

## 教訓

毎日の大半を、PCに向かって行う仕事が多いシステム開発の現場や運用・保守の現場においては、このような「教訓」の唱和をチーム全員で毎日行うことに違和感を持ったり、なかなかそのような時間が取れなかったり、などの状況もあるであろう。

しかしながら、システム障害を未然に防ぐのは、そのシステムに参画しているひとりひとりにかかっている。この事例は、チーム内のコミュニケーションの円滑化とモチベーション向上に役立つ取り組みを行って、過去の障害事例で得た「教訓」を忘れないという大きな成果を生んでいる。

そこで、この事例を紹介することは非常に重要であると考え、この事例のチーム内の個々の「教訓」を取り上げるのではなく、「教訓」共有活動の重要性を取り上げた。

このような活動に取り組む多くの組織が現れることを期待したい。