

1 1-1. プロジェクトの特徴

2 私は銀行業を営む A 社のシステム子会社 B 社に所属す
3 るプロジェクトマネージャである。

4 今回私が論述の対象とするのは A 社の口座振替システ
5 ムのバッチシステムのメインフレームシステムからオー
6 プンシステムへの再構築プロジェクトである。新システ
7 ムの言語は JAVA で OS は Linux だった。

8 本プロジェクトは A 社の勘定系システム再構築プロジ
9 ェクトの一環として実施するものである。他業務のシス
10 テムについても同時期にシステム再構築し、2018年
11 5月に、再構築した各システムを同時に稼働開始するこ
12 ととなっていたため、納期遵守が課せられていた。

13 1-2. リスクにつながる予兆

14 プロジェクトは16年4月にスタートした後、順調に
15 進んだ。しかし、16年12月からの内部設計工程期間
16 中に実施したプログラム製造・単体テストの試行タスク
17 で、担当の稼働時間が計画を大きく超過した。試行対象
18 としたプログラムは仕様がシンプルであり、又、試行担
19 当はオープン系システム開発の経験者であったことから、
20 単純に作業が難航した結果であるとは思えなかった。

21 私はこの稼働時間超過の原因を探るために試行担当に
22 ヒアリングを行った。結果は以下の通りだった。

23 ・プログラムのビルドに多くの時間を費やした。時に
24 は半日費やすこともあった。このため、計画通りの生産
25 性で作業を終わらせることができなかった。

26 ・プログラムコーディング自体はすぐ終わったものの、
27 ビルドの完了がなかなかできないため、単体テストを速
28 やかに実行できなかった。

29 ・単体テストでプログラムバグを発見して修正した際
30 も、プログラム修正はすぐ終わったが、ビルド完了に時
31 間を費やし、修正確認の単体テスト実行が速やかに実行
32 できなかった。

1 2-1. 顕在化すると考えたリスク

2 この兆候をこのまま放置しておくこと、製造・単体テス
3 ト工程に入った後に生産性が計画通りとはならないリス
4 クが顕在化し、作業が遅延してしまうことは目に見えて
5 いると私は考えた。

6 テスト工程に入り、プロジェクトが佳境となっていく
7 中でのスケジュール遅延は致命的であり、2018年5
8 月の納期が遵守できなくなるおそれがあった。

9 なお、本件はプロジェクト計画策定時には洗い出して
10 いなかったリスクであったため、登録簿に追加した。

11 2-2. リスクへの予防処置

12 プロジェクトで導入しているビルドツールはOSSで
13 あり、特定ベンダに全面的支援を願いでることはできな
14 かった。このことから、私は上級マネージャZ氏の許可を
15 得て、我が社の技術統括グループのOSS有識者X氏らに
16 プロジェクトへの参画を依頼した。

17 X氏に調査チームを結成していただき、ビルドの時間
18 が長期化する原因を探ってもらった。

19 調査の結果は、以下の通りであった。

20 ①ビルドツールの設定をデフォルトのままにしている
21 と、ビルドを実行する際に、不要な機能がいくつか自動
22 で稼働してしまっていることがわかった。これら不要機
23 能が動かないように設定を変更すれば、ある程度改善す
24 ることがわかった。

25 ②ビルドに時間がかかるのは、単体テストを実施する
26 PCローカルでビルドするときのみ、それもメモリが少
27 ない旧式のPCでビルドするときのみであることがわか
28 った。例えばメモリが8GBのPCでビルド実施したとこ
29 ろ、ビルドにかかる時間が長くて数分程度となり、大幅
30 に短縮された。また、結合テストや総合テストで使用す
31 るテストサーバでビルドする場合も問題なかった。これ
32 らのことから、PCのスペックが不足していることが、

1 ビルドに時間がかかる原因の一つであることがわかった。

2 なお、導入した OSS ツールのこのような問題は、かな
3 り広く知られている話で、インターネットで検索すると
4 相当数のページがヒットした、とのことであった。

5 調査結果を踏まえ、私は X 氏と対応方法を策定した。

6 まず、不要な機能が稼働しないよう、ビルドツールの
7 設定をデフォルトから変更することをプロジェクト要員
8 に周知徹底することにした (①)。

9 また製造工程以降で使用する予定の PC にいくつか古
10 い型のものがあつたため、マネジメント予備費を使って
11 これらの PC のメモリを 8 GB に増設することにした (②)。

12 2-3. リスクの顕在化に備えて策定した対応計画

13 予防処置が功を奏さずに生産性が高まらなかった場合
14 に備えての対応計画も、私は検討した。

15 前述の通り、結合テストや総合テストでは、試行で発
16 生したものと同様の、ビルド時間が長期化する問題が発
17 生しないことにつき裏がとれていた。このためファスト
18 トラッキングで対応することにした。具体的には、結合
19 テストのテストケース作成・テスト実施を単体テストが
20 終わらずとも、疎通が終了した機能群については、五月
21 雨式に前倒しで開始することにした。

22 ファストトラッキングを実施することにより、作業が
23 手戻るリスクが新たに発生するため、実施する際は新た
24 な重点管理のリスクとして登録簿に追加することにした。

25

26

27

28

29

30

31

32

1 3-1. 予防処置の実施状況と評価

2 17年4月からの製造・単体テスト工程を前にして私
3 は、プロジェクト要員に予防処置を実施してもらった。

4 ビルドツールの設定をデフォルトから変更するにあた
5 っては、X氏ら調査チームにビルド時間を短縮化するツ
6 ール設定の手順書を作成してもらった。プロジェクト要
7 員全員にこの手順書を共有してもらい、全く同じ設定を
8 皆が実施するよう促した。また、古いPCのメモリの増
9 設も前述した通りに実施した。

10 結果として、プログラムのビルド時間は長くて1~2
11 分程度となり、要員から異状があるとの声はあがらなか
12 った。製造・単体テスト工程では、当初計画していた生
13 産性をキープすることができ、予防処置は成功した。

14 リスクは顕在化しなかったため、ファストトラッキング
15 グでの対応で新たなリスクを生むこともなくなった。

16 また最終的に、本プロジェクトは、予定通り18年5
17 月にカットオーバーを迎えることができた。

18 これらのことから予防処置は高く評価できる。

19 3-2. 今後の改善点

20 本プロジェクトでは、対策を打ったのが、リスクが顕
21 在化する直前となってしまった。

22 前述の通り、導入したOSSツールでのビルドに時間が
23 かかるというのは、一般的に知られている話であった。
24 このため、事前に検知できる可能性があったにも関わら
25 ず計画段階ではリスクとして洗い出されていなかった。

26 今後は、計画段階でのリスク洗い出しを改善して、早
27 い段階でリスク管理・対策ができるようにしたい。本プ
28 ロジェクトについてみると、リスク洗い出しにあたりX
29 氏のようなOSS有識者にも話をきけていれば洗い出せ
30 ていた可能性があった。

31 今後プロジェクト計画策定時にリスク洗い出しを実施
32 する時には、広く社内の人間にアンケート方式で登録簿

1 に記入するリスクを募ることを提案してみたい。

2

以上