

オフショア開発能力を高めるには？

オフショア開発能力を高めるには？

中国で、日本からのオフショアプロジェクト開発に対する意見を聞くと、毎回、以下のような意見を多く聞きます。

「仕様あいまいで把握しにくい！」

「コミュニケーションがうまくいかない！」

「納品まで仕様変更が続く！」

日本側に対する不満の声があふれています。

しかし、自分自身の能力に対する検討はほとんどありませんでした。日本企業の信頼を勝ち取るためには、オフショア開発に従業する者の能力を高める必要があります。あるオフショア開発事例をもって検討しましょう。

この案件は日本から発注されたオフショアプロジェクトで、顧客は日本の大手企業である。開発要件には機能もたくさんあり、要求も厳しく、技術と開発ツールも指定されていた。また、顧客側が自主開発したフレームワークを開発とテスト環境で使用する。そのうえ、2 ヶ月半という短い開発期間で納期を厳守しなければならない。いくつもの中国オフショア企業がこのプロジェクトの開発に参加しており、K 社はその一つである。K 社は 40 本の機能仕様の詳細設計、コーディング、単体テスト作業などを担当した。

K 社は、まず開発チーム構成し、ベテランの X 氏をプロジェクトリーダーとした。ただし、X 氏は、この時、初めてリーダーを担当するため、管理経験は不足していた。チームのメンバーは設計チーム、コーディングチームとテストチームを 3 組に分けて、組ごとに 1 人の責任者が指定された。設計チームは 6 人で詳細設計作業を担当した。コーディングチームは 5 人でコーディングを担当し、テストチームは 5 人でテストを担当した。

プロジェクト正式開始 1 週間前に、K 社は顧客先に優秀なブリッジ SE を派遣し、顧客との円滑なコミュニケーションを図れるようにした。プロジェクト正式開始の際に、X 氏は仕様を分析して 40 本の機能を 3 種類に分け、A 種類 10 本、B 種類 12 本、C 種類 18 本とした。この中で A 種類は業務ロジックがほとんどないので、簡単な機能とした。B 種類と C 種類の複雑さはだいたい同じで、機能の業務ドメインは異なっていた。

まず、設計チームは、詳細設計書の作成に取りかかった。同時に、コーディングチームとテストチームのメンバーは、1 週間内に複雑なフレームワークを理解する仕事が割り当てられた。1 週間

オフショア開発能力を高めるには？

経って、部分的な機能の詳細設計書が完了し、厳しい納期守るため、コーディングチームはコーディングを開始した。テストチームもテスト用のチェックリストと試験データの作成を開始した。リーダーの X 氏は、まず簡単な A 種類のコーディングに着手することを計画していた。理由は、簡単な機能を順調に開発し、成功の兆しが見えれば、チームみんなが自信を持つことが出来る考えたからである。

計画にしたがってコーディングチームは、A 種類機能のコーディングを開始した。開発は予想通りに順調進展した。1 本の機能は、だいたい 2 人日で完了となった。1 週間経って、10 本 A 種類機能は全部完了した。テストチームも、この中の 5 本機能の単体テストを完了した。バグ修正した 5 本機能は直ちに顧客の確認を受け、顧客の評価は高いものであった。リーダーの X 氏も、チームメンバーみんな自信に溢れていた。

分析に基づき残った B、C 種類の機能は処理方式によって、入力系とバッチ系と帳票系に分けられた。リーダーの X 氏はスケジュールを調整して、コーディングチームは 1 人あたり入力系 2 本とバッチ系 2 本と帳票系 2 本を担当することとした。設計チームの仕事も予定通りに進捗していた。

B、C 種類の開発がはじまる際に、コーディングチームの仕事には大きな支障が生じた。顧客が提供したフレームワークの複雑さが初めて露見したのである。A 種類の機能は 2 層構成であるが、B、C 種類は複雑な 3 層構成である。どこに値を設定するか、どこから値を取得するか、以前の経験とまるで違ったコーディング方式となることが判明した。この問題を解決するためには、多くの時間が掛かるだけでなく、経験豊富で優れた技術を持つプログラマーが必要であった。

プロジェクトの正式開始から 2 週間経って、作業進捗は 5~10%であった。5~10%という数字は、開発していることを顧客へ表すための数字であった。実際には、ほとんど進捗していなかった。リーダーの X 氏はチームメンバーに残業要求を提出した。

3 週間経って、技術で一番優れた Y 氏の進捗は 40%となった。他のメンバーの進捗は停止状態となっていた。他メンバーは、Y 氏に要領を教えてもらいたいが、Y 氏は「忙しいので、自分で顧客の資料をちゃんと見てください」と言うばかりであった。情緒不安定で、疲労困憊し 1 人のメンバーは病気で休みとなった。2、3 日後には、また 1 人メンバーが病気になって休みとなった。

4 週間経って、10 本 A 種類の機能全部が完了して、顧客の確認を受けたが、B、C 種類の機能は 1 本だけが 80%の進捗率であった。コーディングチームは 2 人が病気で休みのままであった。他のメンバーは、毎日は数百ページの日本語開発資料中に解決策を探していた。リーダーの X 氏は、テストチームの経験あるメンバーをコーディングチームに入れ、コーディング作業の応援をさせた。

オフショア開発能力を高めるには？

5 週間が経って、Y 氏は、ようやく当該機能を開発完了した。ただしプログラムは実行出来なかった。6 週間が経って、プログラムは実行出来たが、いろいろな問題が発生した。テスト結果は予想より悪かった。

納期が迫っていたので、リーダーの X 氏は会社に緊急状況を報告した。他のプロジェクトから 2 人のベテラン SE が支援に来た。詳細設計書の作成がほぼ終わったので、日本よりブリッジ SE も呼び戻された。もちろんチーム全員が残業し続けていた。

フレームワーク知識のシェアリングのため、Y 氏はみんなの開発の技術支援に徹底した。みんなの協力にしたがって、入力系の入力課題とバッチ系の複雑データ処理課題と帳票系の出力課題が一つ一つ解決していった。

納期の直前にコーディングチームの 2 人のメンバーは長期病気休暇から復帰した。幸いにみんなの協力と長期間の残業により、コーディングと単体テストは納品当日に終わった。ただし、単体テストでは多くのバグが発見された。時間がなため、リーダーの X 氏は勝手にチェックリストの確認していないバグの再確認欄に全て「確認済」と記入し、成果物を顧客に提出した。チームは作業を継続していた。

納品 1 週間後、顧客から指摘されたバグ票が提示された。ほとんどが単体テストのバグであった。1 ヶ月後、全てのバグへ対応完了し、チームは解散された。

プロジェクトチームは納期を厳守したが、成果物の品質は保証出来なかった。開発プロセスにおいて、様々な問題が存在していた。以下に分類して列挙する。

スケジュールについて

開発の後期に発生した問題から見ると、プロジェクトのスケジュール作成において、プロジェクトリーダーは適切なタスク分割と実行順序で計画が出来ていない。例えば、B、C 種類ではなく、簡単な A 種類機能を前段で開発している。これでは、B、C 種類の難しさがよく認識できず、開発の進捗を阻害するリスク要因の認識が遅れてしまう。

計画を立てる際に WBS 方法を以ってプロジェクト全体を細かい作業に分割し、クリティカルパスを洗い出すべきであった。クリティカルパスのクリティカルタスクの遅れが、プロジェクト全体のスケジュールの遅れとなることが分かっていたら、B、C 種類はレベルの高いクリティカルタスクとすることができた。

オフショア開発能力を高めるには？

作業を入力系とバッチ系と帳票系に分けて、各々の代表的な機能を選らんで、能力の高い SE が前もって開発サンプルを作成することでリスクの軽減が可能となる。開発サンプルがあれば、開発効率も向上できる。事前準備不十分な場合、問題が発生すると、無駄な高いコストを払うことになる。

開発メンバーの技術能力に応じてタスクを分配を図ることが必要である。簡単なA種類機能の設計と実現は1、2人に担当させ、大部分のメンバー（特に技術能力が高いメンバー）はB種類とC種類機能に配置させるべきであった。

リスク管理について

スケジュールを作成する場合、リスクに対応するための必要な余裕時間と対策を考慮すべきである。問題が発生し、残業で対応することは対策ではない。残業し続ければ、メンバー全員の開発効率は低下する。

プロジェクトリーダーのX氏は、初めてプロジェクトを管理するのであるから、会社は管理面で協力する人を指定するべきであった。

使ったことのないフレームワークに対する、バッチ系など別々の指導的なサンプルは作っていないし、顧客が提供した技術資料も予めよく理解していないし、事前に技術リスクの分析は不足であった。その上、プロジェクトリーダーは「成功の兆し」を望んで、簡単なA種類機能を前もって開発しており、リーダーのリスク意識不足が表面化している。

技術的な問題が発生した場合、有効的な対応をしていなかった。このため、何週間も無駄に費やした。最後にやっと会社に支援を申請するが、時間はもう足りなくなっていた。

報告、連絡、相談について

チーム内部の協力は円滑ではなかった。みんなが適切な方策を見出せないでいる時、Y氏が協力しなかったため、多く無駄な時間が費やされた。

開発期間に余裕がある時に、リーダーは会社の問題を報告していなかったため、会社の支援が遅れてしまった。会社管理者から当該プロジェクトに対するプロセスと品質のチェックも見えなかった。

問題が発生した際に、顧客への連絡もされていなかった。フレームワークは顧客が自主開発した

オフショア開発能力を高めるには？

ものであるら、顧客から支援をもらうことが一番の解決方法であった。

問題が発生しても、チーム内部で知識をシェアしていないし、会社内でもシェアされていない。

開発初期に、1 週間のフレームワーク理解段階があるが、何をしていたのか？ もしこの時に前もって問題が発見されていたら、全体進捗は順調に進めんでいたと思われる。

プロジェクトリーダーが納期を守るためにバグを隠すやり方は大きな間違いである。問題があれば、早期に問題と遅れを報告し、顧客と協力し適切な対策を策定するのみである。悪い品質の成果物を納品することは顧客の信頼を損なうだけである。

立派なプロジェクトをやり遂げるために、いろいろな対策や準備は必要ですが、開発者は前向きにプロジェクト管理能力と技術能力などを高める意識も持たなければいけません。

引用データ URL/引用文献名：

http://www.mypm.net/case/show_case_content.asp?caseID=2710

「対日本软件外包项目失败原因分析」