

## それって技術の仕事！？：“仕様の輻輳”問題 ～心理的輻輳から脱出して健康な開発に～

池田 暁  
クオリティアーツ  
akira.ikeda@quality-arts.com

松尾谷 徹  
デバッグ工学研究所  
matsuodani@gmail.com

### 要旨

本 FP では、多くのエンジニアが経験する「仕様の輻輳問題」を取り上げて議論する。仕様の輻輳を「エンジニア個人に対して仕様に関わる調整事項がエンジニア個人に集中し、開発進捗に大きな影響を生じる状態」と定義した。仕様の輻輳の状態が悪化すると、エンジニアの「心理的輻輳」を引き起こし、それが更に仕様の輻輳を悪化させる。仕様の輻輳は多くの現場でエンジニア個人が解消しようとしているが、本来は PM (Project Manager) の仕事である。べき論はあるが、日々の実際としてエンジニア個人はどう対応すべきだろうか。エンジニア側かつ文化や人間的側面から議論する。

### 1. はじめに

近年、IT プロジェクトの対象が複雑化し、工数ベースの請負作業から、DX など対象分野のスキルや、機械学習/仮想環境のスキルなど、スキル重視ベースへと変化している。高度なスキルを持つエンジニア (コアエンジニアと呼ぶ) パフォーマンスを引き出すことがプロジェクト管理の中で重要な課題となっている。

コアエンジニアがジレンマに陥り、そのパフォーマンス低下を引き起こす仕様の輻輳問題は、実装段階以前にステークホルダー間で調整し、契約や仕様のコミットが必要な様々な要件を決められず、先送りして実装段階へ投げることが主な原因と推測される。我が国独特のあいまいな契約や工程完了のコミット文化が影響している。

文化や組織、人間的な原因を含んでおり、技法や技術だけで解決できない課題であり、かつ状況が深刻化することでプロジェクトや企業の開発競争力にも影響することから、FP のテーマとして取り上げる。今回は、開発現場で実際に生じている問題状況を主にとりあげ、当事者であるエンジニアの側から問題を明らかにし、その後で原因メカニズムのモデル化を試み、解決への糸口について

議論する。

### 2. 仕様に関わる悩み

ソフトウェア開発において、仕様に関わる悩みは多い。仕様定義はもちろんとして、仕様の管理や仕様のプログラムへの変換、テストにおける仕様の取り扱いなど挙げるときりがない。この仕様に関する課題については、仕様記述言語やモデリング手法、仕様のバージョンやトレーサビリティの管理、等に関して様々な手法や提案があり、実用化したり広く普及したりしているものもある。

仕様の不確定や変更から生じる問題という、必ずしも技術だけでは解決できない問題もある。例えば、仕様の実装フェーズにて、顧客をはじめとしたステークホルダーから急な仕様変更依頼が生じたり、それら依頼が複数のステークホルダーから非同期的に生じたり、依頼内容が衝突したりすることが頻発することで、それらを受けたエンジニアには「調整」というアウトプットを生み出さないタスクが増え、本来予定していた進捗に影響を生じるということである。この例のような「仕様の輻輳」が生じると、エンジニア個人はもちろん企業の開発進捗等にも悪影響が生じる。

### 3. 仕様の輻輳とは

本 FP での議論を進めるにあたって、仮ではあるが、「仕様の輻輳」を定義する。

「仕様の輻輳とは、エンジニアのルール外で、納期や仕事量や顧客間で調整が必要な要件変更などを立場上の差 (顧客と請負など) を利用して求め、それら解決が困難なジレンマ状態」とする。輻輳とは物が一か所に集中し混雑することであるが、エンジニアに仕様についての物事が集中することを想起させる言葉として採用している。

仕様の変更は言い換えると仕様の対立である。例とし

て、「顧客対エンジニア」や「ビジネス対開発」がある。この「仕様の対立」は、その「主張者の立場(ロール)」、「仕様に対する観点」、「視座や視点の異なり」等が考えられる。対立があるので解消のための調整が必要となる。

仕様の対立によって生じるエンジニアへの仕様変更依頼が増大していくことで輻輳はより深刻になっていく。

皆さんも、様々な顧客や担当者から、ある仕様項目について定義内容や修正依頼がばらばらに”異なる内容で”なされ、その調整に時間を取られ、開発の進捗も芳しく無くなり、にっちもさっちもいかなかった経験があると思う。そして、おそらく、現在進行系で悩んでいる方も多いと想像する。

#### 4. 仕様の輻輳を引き起こす事象の例

仕様の輻輳問題を考えるにあたって、それを引き起こす例をいくつか挙げてみる。

- 仕様に関する、発注者と受注者での認識が異なっている
- 開発とは同期していない、ビジネス面からのリリース変更や機能追加のさしこみが生じる(確定していた要求が変わる)
- 前工程で要求が固まっていない、(顧客が)誤って承認してしまった等の理由で、あとで要求の変更や追加、優先度付の変更が発生する
- 顧客側の手続きや納期的な面からの調整
- 仕様の実装のために複数の顧客担当者との調整が必要な場合(例えば、サブシステム同士と繋げるためのインタフェース実装)
- ステークホルダーの連携不十分による、異なった、同期しない仕様変更の発生

これらのような事象の発生により、仕様に関する調整依頼連絡がエンジニアにもたらされ、集中することで、仕様の輻輳が発生してしまう。

#### 5. 仕様の輻輳が引き起こす心理的輻輳

仕様の輻輳が発生したとき、その解決や沈静化にエンジニアは、要求や仕様に関する手法を用いたり導入に挑戦したりすることで、事態の改善を図ろうとしてきた。

ただ、実際には、手法のありなしにかかわらず、仕様

変更依頼の量に加え、複数の担当者との調整や仕様の輻輳は、エンジニアに心理的な負担を生じさせ、それが大きくなることでいわば「心理的な輻輳」を引き起こさせる。いわゆる「いっぱいいっぱい」な状態になってしまい、仕様の輻輳を解消するための調整すらも進まなくなってしまう。結果として開発も進まなくなることで、エンジニアはより精神的に疲弊し、ますます事態は悪化していく。担当エンジニアが仕様の調整に忙殺され、疲弊していく姿は皆さんも経験したり目の当たりにしたりしてきたことだろう。

#### 6. 心理的輻輳を引き起こしている状況

エンジニアが心理的輻輳状態になっているとき、どういった状況であるか、いくつか例を挙げる。

- 担当者と、または本来の担当者とは別の担当者との間に忙殺されている
- 直接アクセスできないステークホルダーへの確認の試みが生じている
- ロール外責務の、仕様の取り扱いレベルや実装スケジュール、開発リソースの調整
- 仕様変更解消に向けたプレゼンテーションの準備と実施、フォローアップ

果たして、これらは本来エンジニアの仕事だろうか。

#### 7. 仕様の輻輳を抑えるのは本来 PM の仕事

顧客との、仕様の内容変更や優先度・重要度、スケジュールやリソース配分等の調整は本来 PM が実施すべきことであり、プロセスで解決すべき問題である。だがエンジニア個人は、その生真面目さにより個人技で解決を図ろうとして沼にはまってしまう。

エンジニアは、仕様の対立が生じたときそれを解決するのは PM であるということであらためて理解する必要がある。そして、仕様の輻輳が起きた場合はその解決を PM やステークホルダーに調整や判断を委ねる(提案する)ことが大切である。ただし、委ねるためには、仕様の対立における全体像を理解し、然るべき対策に分け、そのうえで委ねる必要がある。もちろん仕様の輻輳そのものも理解する必要があるし、顧客やPMを含むステークホルダーが仕様の輻輳にどのようなかわり方をしているかも理解する必要がある。さらにはその大きな背景である、国内特有の受発注文化等も理解する必要がある。理解するこ

とが多いため、モデル化等の方法を使い全体理解をしやすくすることが必要であると考え。またモデルがあることでステークホルダー間での使用の輻輳の抑制について議論がしやすくなる。

## 8. まとめ

本 FP では仕様の輻輳および心理的輻輳について議論提起したが、現時点ではあくまで思索にとどまる。本 FP ではエンジニアより「自身や身の回りに起きている仕様の輻輳」について例示いただき、それらが持つ特質や特徴、特に解決すべき課題はなんなのか、複数の事例に共通している課題は何かといったようなことの議論を期待する。

その議論を受けて、今後は、以下のようなことについて議論を深めていきたい。

- 「仕様の輻輳」「心理的輻輳」の議論と定義
- 「仕様の輻輳」「心理的輻輳」のモデル表現
- モデルの活用検討や活用のためのスキル定義
- スキル獲得のための教育
- その他関連する事柄

これらを SS の WG で継続議論をしていく。関心がある方はぜひ議論に加わっていただきたい。