

技術ナレッジ DB システム

－ 「知」の所在を明確にして無駄のない技術支援 －

2014 年 2 月 27 日 初稿

板垣 衛

■ 改訂履歴

| 稿 | 改訂日 | 改訂者 | 改訂内容 |
|----|-----------------|------|------|
| 初稿 | 2014 年 2 月 27 日 | 板垣 衛 | (初稿) |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

■ 目次

| | |
|-----------------------------|---|
| ■ 概略 | 1 |
| ■ 目的 | 1 |
| ■ やりたいこと | 1 |
| ▼ 情報探し | 1 |
| ▼ 登録が面倒じゃないシステム | 2 |
| ▼ イメージは「Twitter」 | 2 |
| ▼ 登録が楽しいシステム | 2 |
| ▼ 反応もまた大事な情報源 | 2 |
| ▼ イメージは「食べログ」 | 2 |
| ▼ イメージは「Yahoo!知恵袋」 | 3 |
| ▼ 増殖する技術の伝道者 | 3 |
| ▼ ユーザー登録前提のシステムによる利便性 | 3 |
| ▼ 連鎖的な技術情報 | 3 |
| ▼ コアな技術の扱い | 4 |
| ▼ 失われた技術の検索 | 4 |
| ▼ 協力会社にも公開 | 4 |
| ■ 要件定義 | 4 |

■ 概略

技術情報をより検索し易くするためのシステムを提案する。

その内容は、要件定義まで至らない、漠然としたシステムイメージの提案である。

■ 目的

本書は、きっちりとした膨大なドキュメントを管理するものではなく、もっと手軽に「知」（＝ナレッジ：Knowledge）の「所在」を明確にして、技術を習得するための「手がかり」を得ることを目的とする。

また、これにより、ドキュメント作成にかかる労力の削減、プロジェクトにまたがって利用価値の高い技術の検出なども目的とする。

■ 【提案】このシステムでやりたいこと

提案として、このシステムで「何をやりたいのか？」を示す。

▼ 情報探し

開発の最中、技術情報を検索したい時に考えることはどんなことか？

- ・ 「プログラムのこの技術について知りたいんだけど、誰が詳しいだろうか？」
- ・ 「DCC ツールのあの技術を習得しようと思うのだけど、どの書籍が最適だろうか？」
- ・ 「内製ツールを使っていてなんかうまくいかないんだけど、誰かわかる人は解決方法を教えてほしい」

本システムが目指すのは、「そこに技術情報があるシステム」ではなく、「誰に聞けばいいのか？」「どの書籍がいいのか？」「わかる人が質問に答える」、そんなシステムである。

プログラマーなら、「このプログラムに書いてあるから見ておいて」なんて言いたい場面はしょっちゅうである。とにかく、なるべく面倒をかけずに、「どんな情報がどこにあるか？」を明確にすることを重視する。「細かい説明は人に聞けばいい」という方針である。

▼ 登録が面倒じゃないシステム

とにかく、情報の登録が面倒なシステムは誰も書き込まず、結局利用価値のないシステムとしてすたれてしまう。そうならないためにも、手軽に、少ない情報の登録で済むものとする。

「こんなの作ったんだけど、知りたい人は聞きに来てね」程度の情報を書き込んで公開するシステムである。

▼ イメージは「Twitter」

情報の登録は、Twitter なみに簡潔に済むものとする。

▼ 登録が楽しいシステム

一方的な登録ではなく、Facebook の「いいね！」や Amazon の「役に立った」のような反応が得られるシステムとする。

それにより、情報を公開する意欲につなげる。

ちなみに、「悪い」評価は直接記入せず、メールを出す形を取り、投稿を悪評にさらして投稿者が以後気後れするようなことにならないようにする。

▼ 反応もまた大事な情報源

情報を検索する際、「どんな反応があったか？」は重要な指標となる。

コメントを通した追加情報のやり取りを重ね、やがて、より洗練された情報へと成長する。

▼ イメージは「食べログ」

食べログの「お店探し」のような「技術探し」が本システムの目指すスタイルである。

食べログはお店ごとの評価やコメントが確認できるほか、細かい検索条件を設定する前から注目度の高いものを提示してくれたり、関連する情報を合わせて表示してくれたりもする。

技術情報も、「この技術を検索したい」という要件に対応するだけでは、「検索すべき技術を知っている」ことが前提となってしまう。有効な技術をより広範囲に提示するためにも、

直接求められていない技術情報であっても、議論が活性化し、評価の高い技術は、食べログのように一方的に見せつけるようにする。

食べログにおける「地域」の設定と同様に、技術カテゴリ（「プログラム」や「DCC ツール」など）の指定や、さらにそのサブカテゴリを段階的に指定できるものとし、その過程で、そのカテゴリで注目度の高い情報をいっしょに表示していくようにする。

▼ イメージは「Yahoo!知恵袋」

技術的な質問を受けたらそれに答える。この当たり前のやり取りがこのシステム上で行われれば、それだけで有益な情報として蓄積される。

本システムは、「Yahoo!知恵袋」のような、「誰か」に助けを求める Q&A システムも実装する。

▼ 増殖する技術の伝道者

技術の評価は「いいね!」「役に立った」だけではない。「理解した」（この件は僕に聞いてくれてもいいですよ）という評価も行えるものとする。これにより、検索で得られる「知」の「所在」を増やしていけるようにする。

▼ ユーザー登録前提のシステムによる利便性

本システムの特徴の一つとして、社内システムということもあり、「匿名性がない」ことがある。

情報を投稿したユーザーやそれに反応したユーザーは全てわかる状態で動作するため、「その情報や質問の発信者は誰か？」は記入しなくても確実に登録される。

また、自分が登録した情報に対する未読のレスポンスはまとめてリストアップされるものとして、迅速な反応ができるシステムとする。

▼ 連鎖的な技術情報

「知」の登録、「質問」の登録は、極めて自由度が高く、あらゆる情報から派生できるものとする。

例えば、A さんが登録した技術情報に B さんがコメントをつけ、そのコメントに対して C さんが Q&A 形式の質問を出す、といった流れを可能とする。

また、関連情報も明示的に「関連情報」として登録することで、直接検索にヒットしな

かった情報も検索結果に出すことができるようになる。

Amazon や AppStore のようにコメントがずらずら続くだけのシステムではなく、連鎖的に情報を扱っていくようにする。

▼ コアな技術の扱い

社内システムであるため、社内に特化した技術情報もあつかっていく。

▼ 失われた技術の検索

契約満了などに伴い、開発現場を離れるスタッフしか持ちえない技術を明確にし、技術の継承を促したいような場面でもこのシステムは活用できる。

▼ 協力会社にも公開

開発を請け負う協力会社にも本システムを公開し、積極的に利用してもらう。

ある程度は「開発タイトル」などのカテゴリに基づいて情報をマスクするものの、できるだけ多くの人がこのシステムを共有することによる利便性の方を重視する。

■ 要件定義

提案に賛同が得られたなら、以上に内容に基づいて、システム要件の定義や、画面イメージなどをまとめていく。

■■以上■■

■ 索引

索引項目が見つかりません。

技術ナレッジ DB システム

以 上