

～思考プロセスがなぜ必要なのか～

《実技》システム開発 思考プロセスのご案内

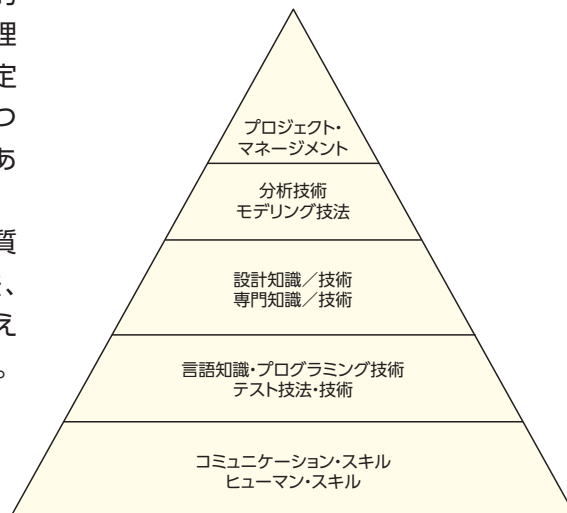


思考プロセスとは

システム開発の「思考プロセス」は、実務では「知識や技術・技法」をただ覚えて使うのではなく、その本質や目的などを正しく理解し、設計や実装の理由や根拠、目的やポリシーを正しく定義し成果物に反映する事で品質につながる重要性を身につける必要があります。

現場で数年掛けて身につける品質に対する意識や考え方、技術や技法、設計の本質など優秀な有識者の考え方やノウハウが「思考プロセス」です。

<思考プロセスと知識や技術スキルの関係>



思考プロセス			
導入編	実装編	設計編	分析編
新入社員	製造工程担当(新入社員・2年目以降)	プロジェクトリーダー(2～3年目以降)	プロジェクトリーダー(3～4年目以降)

社員スキルの現状

昔から社員教育は基礎的な教育のみで、「本質的な教育は現場で実践しながら身体で覚える」というのが一般的です。確かに理に適った現実的な教育であると思いますが、昨今では昔のようにじっくり時間と人材をかけて開発することは稀となっています。

さらに、一時期の不況から優秀な中堅社員が多く他業界に流出したこともあり、現場で新人から中堅社員を教育できる(教育の仕方を熟知している)中堅社員が少なく空洞化が進んでいるように感じます。

また、社員教育においても「研修は基礎的な知識や技術を習得すれば良い」という考えが変わっておらず、多くの優秀な人材を育てることが難しくなってきました。

最近では考える教育(直ぐに答えを教えずに受講者に考えさせる)も当たり前となってきましたが、まだまだ足りず

一番大切な教育がされていないと感じています。

もっと知識や技術にとどまらない、本質的な教育(熟練者の思考や判断基準、品質に対する姿勢や対応)が必要であると感じます。

- ◎なかなか成長しない
- ◎異なる作業や成果物で同じ失敗をする
- ◎指示待ち / 他責にする
- ◎自分で評価や判断ができない
- ◎安心して仕事を任せられない
- ◎フィードバックができない
- ◎品質を上げるための具体策をうてない

なぜ品質が良くなるらないのか

どの企業でも品質に問題を抱えています。「何故品質が良くなるらないのか?」永遠のテーマではありますが、前述の通り当たり前の結果と言わざるを得ません。

新人から中堅社員において、品質に対する意識が低いことが挙げられます。

「品質」を重視しなければならないことは「皆がわかっています」が、品質を高くするために具体的に「何が重要(必要)か」を正しく理解できていない方が多いからです。

例えば、設計書のレビューで「この設計になった理由は?」と聞くと「要件からこの様な設計になりました」と答えます。

また、「品質を確保するために必要なことは?」と問うと「テスト」と答えます。

極端ですが「テストをすれば品質が保てる」「設計書を作れば品質が保てる」と考えている社員は決して少なくありません。

また設計するのではなく、設計書を完成することが目的となってしまう社員も多く存在します。

この様に品質にとって「何が重要であるか?」「何を考え・議論検討し・決断するか」といった本質的な教育が必要です。

思考プロセスができないと

「思考プロセス」はこの様な技術や知識ではなく、熟練者の「考え方」を教育するために開発されたコースです。思考プロセスができない社員の現場でよく起こることは…

分析工程では…

- 与えられた資料(お客様からの「要求仕様書」や設計書)の読み込みが浅い
- 書かれている文章を読んでいるだけ
- 部分的に理解・認識して終わりにになっている
- 成果物が曖昧であったり、抽象的になる
- 書かれている事や求められている事を鵜呑みにする
- 全てをそのまま実現しようとする
- 書かれていることを実現すれば良い製品になると思っている
- 技法に則り成果物を作ること満足して(分析した気になって)しまう
- 成果物間に関連性や一連性がない
- 要件分析と要件定義を混同している
- 重要な用語(業務用語や一般用語)の定義がされていない、または曖昧
- お客様やメンバー間で意識のずれが発生する

- 全体を意識していない、または長期的な運用で何が起こるか考慮していない
- 分析しながら要求を確定(都度懸案を解決)してしまう
- 懸案について分析されていない
- 懸案が全体に対してどの様に関わり影響を与えるか考慮されない
- 要件がなかなか確定できない、懸案が下流に入っても解決しない
- 分析が浅くなり本質や見えない事(懸案事項)が認識できない
- アーキテクチャや運用がその場凌ぎになる
- 下流工程でコストが膨らむ(プロセス/プロダクト品質が低下する)
- 要件や仕様の漏れ
- 要件や仕様の粗漏
- 仕様変更や仕様追加
- 試験漏れ
- 不具合

その他全般では…

- 要件や設計、作業などで曖昧な認識のまま着手する(直ぐに成果物を作ろうとする)
- どの工程においても、提供物(要件や設計・作業手順など)に書かれていないことや見えないことを発見できない(結果「漏れ」となる)
- 目的を見失った作業や行動(成果物の完成や作業することが目的になってしまう)
- 関連した作業の孤立(前後の成果物や作業について関連性や影響などを理解・意識できていない)
- 不必要な議論や検討による時間の浪費

- 計画性の無い作業や行動
- 想定外の何かが発生する(リスクの検討が足りない・できていない)
- 思いつきや勘(理由や根拠がない)
- 他人に説明できない
- 会議などで、的を得ていない議論が続く・議論が収束しない/発散する
- できないことばかり並べて、対策や対応・代替案を立案できない
- 思い込みや誤った認識や理解
- 連絡や報告を怠ることによる不具合
- 他責や無責任な言動

など、プロセス品質⇒プロダクト品質に直接影響する事例は数え上げればキリがありません。

思考プロセスができれば

例1) 思考プロセスを徹底的に教えこんだAさん(社会人2年目の新人)との会話

Aさん:「他社の中堅社員と議論していると、自社の新人と話しているように感じる…」

講師:「どうして?」

Aさん:「議論の論点はズレるし、要件や設計の目的や観点・方針が全く考えられていないから、議論にならない…」

例2) 同様に設計～実装を外部に発注したBさんの話

Bさん:「思考プロセスが役に立ってます。」

講師:「どんなことが?」

Bさん:「この前、発注している協力会社との設計～実装のレビューで、設計と実装に根拠やポリシーが全く無くて酷かった…」

例3) 同様に要件分析を任されているCさんの話

Cさん:「思考プロセスを受講する前は、自分の分析や設計に全く自信が持てなかったし、判らないことが多すぎて悩んでばかりで大変だったけど、最近は(受講してからは)不安を感じなくなりました」

講師:「見えないことが無くなったでしょ!？」

Cさん:「はい、なんで今までできなかったのか不思議です。思考プロセスで考えればどんな工程でも効率良く最善の選択ができるんだなって、改めて感じました…」

思考プロセスでは一般的に教育する「知識や技術・技法」を教えるのではなく、それらの技術や技法をどの様に使うのか(なぜ必要なのか)、「技術や技法」は先人が考えた「品質を保つためのノウハウの結晶」です。その本質を理解し、現状のプロダクトに上手に適応する必要があります。

技術や技法を使い設計するのではなく、理由や根拠、目的やポリシーを持ち分析・設計・実装から試験まで行い、現状にあった最善の選択をすることで、品質につなげる考え方を教えています。

ただ「論理的に考える・行動する」ではなく、論理的に考え・行動するために必要な「目的・根拠・観点・方針など」を常に意識(明確に定義)する重要性を理解し、具体的に品質に影響する意識改革を行います。

思考プロセスを受講すれば、次の日からできるようになるわけではありませんが、前述したような事象は少なくなっていく事は確かです。(過去の受講者の意見から)

思考プロセスにより、品質を確保するために具体的に何を考え分析～設計・実装する必要があるかを理解し実践できるようになるでしょう

できれば、単独で「思考プロセス(導入編・実装編・設計編・分析編)」だけを教育するのではなく、全てのカリキュラムに対し「思考プロセス」の考え方で教育することで、知識や技術・技法の本質や何故必要なのか、メリット/デメリットなどを理解し、必要な時に必要なレベルで、適用ではなく適応できる人材を育成できると、理想だと考えております。

「自分の仕事に自信と誇りを持ち、ポリシーとプライドのある(品質の高い)成果物を残せる優秀な人材」が増えるように…

品質とは<品質モデル(特性)>

品質とは何でしょう? 次に一般的な品質特性を上げます。(詳細は省きます)

<品質モデル(特性)>

X0129-1:2003(ISO/IEC 9126-1:2001)より抜粋

主要品質特性	外部品質および内部品質					
	機能性	信頼性	使用性	効率性	保守性	移植性
副品質特性	合目的性 正確性 相互運用性 セキュリティ	成熟性 障害許容性 回復性	理解性 習得性 運用性 魅力性	時間的効率性 資源的効率性	解析性 変更性 安定性 試験性	環境適応性 設置性 共存性 置換性
要求	機能要件		非機能要件			

簡単に言うと

- お客様の要求通りで正確な結果を返す製品を作る
- 不測の事態が発生しても障害を最小限に留め、障害発生前の状態に戻せる製品を作る
- 機能の追加・変更に対し、誰でも簡単に実現でき、その影響範囲が最小限になる製品を作る
- 製品の品質が高いことを証明(検証)しやすい製品を作る
- お客様にとって解り易く簡単で、特殊な知識や技術を必要としないで使える製品を作る

です。これらの製品に関する品質のことを「プロダクト品質」と言います。

既存社員向け研修 概要

●思考プロセス(導入編) ～論理思考とSE基本動作

コード T-1

講座日数/0日(新人向けには1日/中堅社員にはテキストの事前配布で対応)

講座時間/9:00～17:30(最終18:00または20:00)

対象者/○新入社員 ○プログラマ ○システムエンジニア

コース概要/

○仕事をする上で最も必要な、論理思考と思考に沿った行動に必要な要素を学びます。

○熟練者が何故スマートに仕事を熟せるのか? 熟練者が身に付けている考え方やチーム活動でのポイントについて、初心者が陥りやすい失敗例からわかりやすく学びます。

○基本動作や確認(報告・連絡・相談)、計画に沿った行動の重要性を学びます。

到達目標/

○論理的な思考と思考に沿った行動に必要な要素を理解できる。

○仕事をスムーズに進めるための考え方や基本動作を理解できる。

○確認(報告・連絡・相談)の重要性を理解できる。

○スケジュールに沿った行動の重要性を理解できる。

コース詳細/

I. SEにとって重要な要素

II. 思考プロセス

1. 論理思考

1) 目的を明確にする 2) 前提条件(理由)を明確にする

3) 結論を導く、観点(考え方)を明確にする

4) 方針を明確にする 5) 効率的な作業手順を明確にする

6) やりながら、進め方を考える

7) 常にイメージを持ち、イメージのネットワークを作る

8) フィードバックを必ず行い『思考プロセス』の質を上げる

2. 基本動作

1) マナー 2) チーム活動 3) PDCA 4) 健康管理

3. 意欲と心構え

4. 知識と技術

※中堅社員向けコースでは、テキストの事前配布で、次コースでポイントを簡単に解説します。

●思考プロセス(実装編) ～拡張性と保守性のあるプログラミング(C言語版/Java版)

コード T-2

講座日数/2日間

講座時間/9:00～17:30(最終18:00または20:00)

対象者/○新入社員 ○プログラマ ○システムエンジニア

コース概要/

○製造工程における品質の重要性を理解し、拡張性と保守性のある実装設計およびプログラミングに必要な要素技術と考え方を学びます。

到達目標/

○拡張性と保守性の重要性を理解できる。

○実装設計の重要性を理解できる。

○拡張性と保守性のある実装設計に必要な基本技術を身につけます。

○拡張性と保守性のあるプログラミングスタイルを身につけます。

コース詳細/

<1日目>

演習課題1. 悪い設計とプログラムの保守(改修)が、どれほど大変か知る
バグのあるプログラムの改修(C言語版/Java版)

I. システム開発

1. 論理思考 2. 品質とは 3. システム開発標準(工程)とは

4. 設計(設計書)とは

II. ロジック構築

1. ロジックとは 2. 演習課題の問題点(プログラムと設計書について)

3. モジュール設計での思考プロセス

<2日目>

III. アルゴリズム(定石や各種パターン)

IV. コーディング・スタイル

1. コーディング・スタイルの一例 2. 必ず守って欲しいコーディング・スタイル

V. プログラム構造設計

1. プログラムとは 2. データ構造とは 3. データ構造からプログラム構造を作る

4. モジュール(関数やメソッド、クラス)分割の観点

5. プログラム設計書に書くべきこと

演習課題2. カレンダープログラムの構造設計

正しいプログラム構造を設計してみる

●思考プロセス(設計編) ～目的と方針のある設計

コード T-3

講座日数/2日間

講座時間/9:00～17:30(最終18:00または20:00)

対象者/○プログラマ ○システムエンジニア ○プロジェクトリーダー

前提条件/ 思考プロセス(実装編)を受講済みであること

コース概要/

○設計とは何か? 設計書に何をどこまで記載すべきか?

○設計の本質を理解し設計の品質を確保することの重要性を学びます。

○成果物のレビューや評価を自己で確認/判断する要素を学びます。

到達目標/

○設計書やドキュメントの重要性と本質を理解できる。

○設計書やドキュメントの目的や方針の重要性を理解できる。

○目的や方針のある設計・ドキュメントを作成できる。

○目的や方針から、成果物の品質を自ら確認および向上できる。

コース詳細/

<1日目>

I. システム開発

1. 要件とは 2. 設計とは 3. 良い設計にするために 4. 品質とは

II. 思考プロセスの設計への適応

1. 目的を明確にする 2. 前提条件(背景や理由)を明確にする

3. 方針を明確にする 4. リスクを検討する

5. 設計品質の評価観点を定義する

演習課題1. ①(講義)業務プロトコル設計(入門)とは

②グループ演習(簡単なプロトコル設計)

③演習の解説(設計の目的や方針を理解する)

<2日目>

演習課題2. ①(講義)コンポーネント設計(入門)とは

②グループ演習(簡単なコンポーネント設計)

③演習の解説(設計の目的や方針を理解する)

演習課題3. ①(講義)システム全体の例外設計(入門)とは

②グループ演習(簡単な例外設計)

③演習の解説(設計の目的や方針を理解する)

●思考プロセス(分析編) ～要件を分析し把握する技術

コード T-4

講座日数/2日間

講座時間/9:00～17:30(最終18:00または20:00)

対象者/○システムエンジニア ○プロジェクトリーダー ○プロジェクトマネージャ

前提条件/○思考プロセス(設計編)を受講済みであること

○UML(入門レベル)の知識があること

コース概要/

○要件分析の3つのモデリング技術(入門)を学びます。

○要件分析工程の進め方を理解します。

○見えない事(要件に書かれていない重要事項)を見抜く重要性を理解します。

到達目標/

○要件とは何かを理解できる。 ○要件分析の勘所を理解できる。

○要件分析技法(概念データモデリング・業務モデリング・ビジネスルールモデリング)の入門レベルを理解できる。

○技法の限界を知り、正しく使う技術を身につける。(技法に振り回されたり、技法を使うことで満足しない)

○複数の技法から「見えないこと」を見抜く力を身につける。

コース詳細/

<1日目>

I. 要件とは

1. 要件の落とし穴 2. 品質とは 3. 上流工程で起こしやすいミス

4. 要件定義…1)フェーズ1.要件を把握する 2)フェーズ2.見えないことを整理する 3)フェーズ3.要件を定義(確定)する)

II. 要件分析技法(入門レベル)

1. (講義)概念データモデリングとは…個人課題1.概念データモデリング

2. (講義)業務モデリングとは…個人課題2.業務モデリング

3. (講義)ビジネスルールモデリングとは…個人課題3.ビジネスルールモデリング

<2日目>

演習課題1. 要件を把握する ①グループ演習(簡単なシステムの運用改善提案)

演習課題2. 各モデルから業務仕様を整理する ①個人演習

演習課題3. モデルを要件から評価する ①個人演習

お問い合わせ先 パンフレット内容でご不明な点やご質問・ご相談(受講料・研修費用)がございましたらお気軽にお問い合わせください。



株式会社 日本教育情報センター

〒183-0023 東京都府中市宮町2-15-13 第15三ツ木ビル

TEL:042-336-5311 FAX:042-336-5314

http://www.socialskill.jp/ E-MAIL:recep@socialskill.jp