

クオリティ ワン  
*Quality One*

Vol.10 2010 年 5 月号

Software Quality Profession

財団法人 日本科学技術連盟

## 1. 品質

### ■ 派生開発の混乱を救う「XDDP」

( eXtreme Derivative Development Process )

株式会社システムクリエイツ  
代表取締役 清水 吉男

#### 1. 派生開発の問題

派生開発というのは、ある製品やシステム(のソースコード)をベースにして、新しい機能を追加したり、操作性などを改良して製品やシステムを作り上げて行く開発方法で、その過程で新しい製品やシステムの体系が生み出されることがあります。

派生開発のプロジェクトの特徴として、一人プロジェクトになるような小さな変更案件から数十人で対応するような大きな変更案件まであることです。開発期間が1ヶ月という短期間のものから1年以上のケースもあり、変更行数が500行程度のものから10万行に達するようなものまであります。

また、多くの場合、ソースコード上で該当すると思われる箇所を見つけ次第に変更しています。もともと変更要件の実現方法も複数あり、いずれの変更方法でもテストでは「正しい」と評価される可能性があります。でも担当者が気づくのは通常はその中の「1つ」です。しかも必ずしも適切な変更箇所が選ばれているとは限りません。

また、プロジェクト終了時の反省会では、相変わらず「全体を理解できていなかったから…」という声が出てきます。そしてそれで皆さんが納得しているのです。

でも

- 「全体」とはどの範囲ですか？
- 「理解」とはどの状態を言うのですか？

要するに「全体」を理解して作業ができる状況にないにもかかわらず、「全体」を理解すれば問題が解決すると考えているのです。そしてこの立場に立つ限り、全体を理解できない状況に備えることができません。それよりも最初から「部分理解」の制約の中で作業が強いられるということを前提にすることで、担当者の思い込みや勘違いに気づく方法を取り入れる工夫ができるのです。

また、派生開発では下表のように新しい機能の追加に伴って問題が発生しているケースが少なくありません。これも派生開発の特徴です。

変更のタイプ		「変更」で実現する部分	「追加」で実現する部分
仕様変更	変更 追加 削除	機能レベルで変更仕様(変更要件)が書かれたものがあり、それを元にソースコードの該当箇所を探して変更する	
	機能単位に新規作成	追加／流用(移植)で実現するソースコードを受け入れるためには、ベースのソースコードを変更しなければなりません	追加機能の要求仕様書(要件定義書)が書かれていて、それを「設計」し「実装して実現します
機能追加	他システムから流用(移植を含む)	その部分を変更仕様(変更要件)として書き出されたものではありません	他のシステムからソースコードを切り出して、ベース(受け入れる側)のソースコードに埋め込みます

#### 2. 保守開発との違い

もともとソフトウェアエンジニアリングの世界には、「保守」あるいは「保守開発」というプロセスがあります。(JIS X 0160 : ソフトウェアライフサイクルプロセス)

保守のタイプ		作業内容
訂正	是正保守	ソフトウェア製品の引き渡し後に発見された問題を訂正するために行う受身の修正。
	予防保守	引き渡し後のソフトウェア製品の潜在的な障害が運用障害になる前に発見し、是正するための修正。
	緊急保守	是正保守実施までシステム運用を確保するための、計画外で一時的な修正。
改良	適応保守	引き渡し後、変化した又は変化している環境において、ソフトウェア製品を使用できるように保ち続けるために実施するソフトウェア製品の修正。
	完全化保守	引き渡し後のソフトウェア製品の潜在的な障害が、故障として現れる前に、検出し訂正するための修正。
	改良保守	新しい要求を満たすための既存のソフトウェア製品への修正(機能追加を含む)。

2008年に「緊急保守」と「改良保守」が追加された。参照 <http://www.jisc.go.jp/>

「保守開発」で想定しているのは、主にバグなどの「是正保守」や稼働後の環境の変化に対応するための「適応保守」であり、そのようなケースでは今でも「保守」のプロセスで対応しても支障はないと思います。

ビジネス分野においては、このようなケースが多いと思われそうですが、それでも最近では、新しい機能の追加なども伴うことが多く、変更要件が複雑になっていると思われまます。

これに対して、組み込みシステムの世界では「是正保守」よりも、機能追加や性能の改良などを行って新しい製品として市場に出すというケースが多く、そこには「保守開発」という概念はありません。たとえば「携帯電話」が今日の「ケータイ」端末に変化(進化)する過程は「保守」では説明がつきません。パッケージソフトや流通などの制御システムでも、ビジネスの競争に勝つという要求に応えるために頻りに機能の追加が行われています。

こうした中で、2008年にJIS X 0160が改訂され「緊急保守」と「改良保守」が追加されました。「改良保守」は主に機能追加など新しい要求への対応を想定しています。ただし、「是正保守」と「改良保守」では求められている「要求」が根本的に異なるという、新しい問題が生じます。

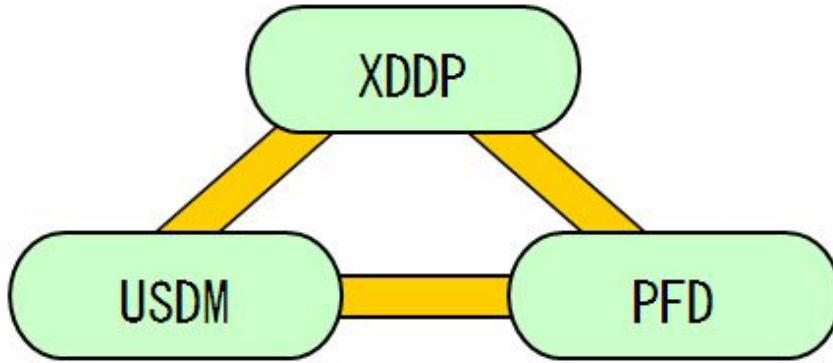
### 3. XDDP の考え方

「XDDP」は派生開発に特化した開発アプローチで、派生開発の特徴である「短納期」や「部分理解」の中での作業に対応しています。短納期の要求に対しては無駄のない合理的なプロセスで対応する必要があります。また「部分理解」に対してはいきなりソースコードを変更するのではなく、適切な成果物を生成して他の人のレビューを受ける機会を作ることが重要です。

「部分理解」に対応した開発アプローチによって手戻り作業が大幅に減り、結果として「短納期」を支援することにもなります。XDDPによって「Q」と「C」「D」を同時に手に入れることに繋がるのです。

また派生開発では、変更行数が少ないときは「一人プロジェクト」になるのはやむを得ませんが、「部分理解」の下で生じる思い込みや勘違いを防ぐ方法を講じる必要があります。XDDPの開発アプローチはこの問題にも対応できます。

「XDDP」は、下図のように「USDM」と「PFD」の2つの主要な技術に支えられています。



「PFD」(Process Flow Diagram)は、今回の派生開発の要求(納期やコストも含む)を達成するために合理的な成果物とプロセスの連鎖を設計する技術であり、これによって手戻りが起きにくい開発アプローチを設計します。また途中での変化に対しても、開発アプローチの「別案」を考え出すための重要なツールなのです。

ただし、設計された開発アプローチの合理性を判断するには、そこに現れる成果物の構成などの定義とプロセスにおける作業の定義が必要になります。

一方「USDM」(Universal Specification Describing Manner)の方は、今回の要求(追加/変更)を適切に表現し、そこから必要な仕様を引き出す方法を提供します。USDMは要求と仕様を階層関係の中で捉えますが、これはもともとモレが生じにくい構成です。そして要求に含まれる「動詞」をしっかりと表現することで、要求に含まれる「仕様」を引き出すテクニックを提供しています。

このようなUSDMの表記法によって、実際にベースライン設定後の仕様変更率が「1%」にまで押さえることもできています。

#### 4. XDDPの特徴

XDDPの主な特徴を以下に列記します。

- 変更と機能追加をそれぞれ別々のプロセスで対応し、これらを並行させます
- そのため、変更要求仕様書と追加機能要求仕様書の2種類の要求仕様書を用意します
- 変更と追加の要求仕様書は「USDM」の表記法を用います
- 変更プロセスでは変更要求仕様書、TM(Traceability Matrix)、変更設計書の「3点セット」の成果物を作ります
- 変更要求仕様書や変更設計書はすべて「before/after」で記述します
- これによって、追加も含めてXDDPで生成する成果物はすべて「差分」になります
- 機能追加は追加機能要求仕様書を作成すると同時に、それをベースのソースコードに受け入れるための変更要求仕様書に対応させます
- ソースコードの変更は、すべての変更箇所と変更方法を明らかにしたあとで一斉に取り掛かります
- 機能仕様書や設計書などの公式文書は、テスト終了後に「差分」の情報によって更新します

もっとも重要なことは、「変更」と「追加」を異なるプロセスで対応することです。「追加」は通常の要求仕様書が必要で、その後の設計プロセスや実装プロセスも新規開発のアプローチが使えます。そこに現れないのはシステム全体の「アーキテクチャを設計する」プロセスぐらいです。

一方「変更」は、簡単なケースでは変更要件から該当箇所の当たりをつけてソースコード上で見つけて変更する、というプロセスも必ずしも否定できません。税率の変更などはこのような対応でも問題は生じないでしょう。

もちろん、変更要件の内容によっては、もっと緻密なプロセスが必要になります。たとえば現状のソースコードを解析して不足している設計書の記述を補いながら、関係箇所の相互関係の理解を進めたり、メモリの競合やCPUの負荷の状況を検証したりしながら、最適な変更箇所を特定するといった方法も必要です。しかも、こうして見つけた「変更箇所」も担当者の思い込みや勘違いの可能性があり、レビューの機会を設けながら作業を進める必要があります。

XDDPでは、このように「追加」と「変更」の要求が異なることを考慮し、変更プロセスにおいて「変更要求仕様書」

「TM (Traceability Matrix )」「変更設計書」の3種類の成果物(3点セット)の作成を勧めています。これらは「What」「Where」「How」の3つの視点から変更箇所の情報を表現しています。

「変更要求仕様書」は、変更要求に対して変更箇所を「変更仕様」として記述しますが、すべて「before / after」で表現することを求めています。「追加する」や「削除する」は、それ自身が「before / after」を内包している用語です。そこで XDDP では「変更」も「現状の仕様」に対して「新しい仕様」を対峙して記述することで「変更」の意図を表現します。この副次効果として影響箇所の気づきを誘導し、変更の難易度や変更行数の見積もりも容易になります。

「3点セット」のポイントは、生成するタイミングが異なることです。従来は、変更要件ごとに

- ①該当箇所(変更箇所)を見つけて
- ②変更方法を考え
- ③その場でソースコードを変更する

という一連の作業が行われますが、XDDP では①で変更要求仕様(と TM)を作成し、すべての変更要件に対して①の作業が終わった後にレビューを行い、担当者の思い込みや勘違いを除去してから②の作業に入って変更設計書を作成します。そしてすべての変更仕様に対して変更設計書を作成し必要なレビューを終えてから、変更設計書に沿ってソースコードを一気に変更します。

このように「3点セット」の成果物を作るタイミングが工夫されていることで、担当者自身でも思い込みや勘違いを除去できるのと同時に、それぞれの段階で関係者によるレビューを可能にしているのです。そして最後になって一気にソースコードの変更に取り掛かります。この一気呵成のソースコードの変更は XDDP の特徴の一つです。

こうすることで、複数の変更要件で変更箇所が重なるケースにも対応しやすくなり、従来のように何度もソースコードを変更し直す必要もなくなります。また別の変更要件の探索中に、既に対応した変更要件の対応方法で漏れていた箇所などに気づくこともあります。その場合でもまだソースコードを変更していませんので、より適切な情報に基づいて変更要求仕様を訂正できるのです。

## 5. XDDP の効果

XDDP の開発アプローチは適切な成果物を生成しながら作業を進めるため、途中で思い込みや勘違いに気づき易く、総じてバグは減ります。特に、複数人数で対応するケースでは、従来では各担当者が分担するソースコードを結合してテストに入ったところで、お互いの食い違いが元でバグが噴き出します。XDDP では「3点セット」の成果物によって互いに変更箇所やその意図を確認する方法を提供してことで、テストで発見されるバグは大幅に減るのです。

また「一気にソースコードを変更する」というのも XDDP の特徴です。一般に上位のプロセスは要件の内容が生産性に影響を与えますが、実装プロセスは事前の準備さえ整えておけば個人の生産性データがそのまま適用できます。要件の内容に左右されることはありません。

XDDP では変更設計書によって、すべての変更要件について具体的な変更方法まで記述していますので、ソースコードの実装工程は予定通りに進行します。ソースコードの同じ箇所を何度も変更することはありません。

従来の方法ではソースコードの変更に着手するのが早く、1時間あたりのコード生産性が「5行前後」というケースも少なくありません。その結果、20000行の変更(追加も含む)に4000時間(5人で4ヶ月)必要になります。

これに対して XDDP ではコード生産性は一般に80行~100行に達します。

	コード生産性	所要時間	一人あたり実装時間
従来方法	5行	4000時間	800時間(4ヶ月)
XDDP	80行	250時間	50時間(1週間)

有効変更行数=20000行

担当人数 =5人

この結果、従来方法ではソースコードの変更作業を終えるのに合計5000時間超の工数が必要になります。ま

たその後テストに回されたところで200件を超えるバグ(10件/KLOC)が発生します。さらに変更箇所の記録がないためにバグの対応に1000時間超の手戻り工数(平均5時間/1件)が費やされます。

これに対して、XDDPでは「3点セット」の成果物を作りながら作業を進めますので、ソースコードの変更作業はわずか250時間、日程にして約1週間で終わってしまいます。つまり、3750時間(4000時間-250時間)を投入して「3点セット」の成果物を作ってレビューすれば、わずか1週間余りでソースコードを変更できるということです。

実際には変更依頼から変更設計書に展開するのに「3750時間」を必要とせず、2割~3割ぐらいは余ります。この「2割」というのは従来方法では

- 変更作業を進める中で変更要件を依頼者に確認しなおしたり
- 何度もソースコードの同じ箇所を読み返したり
- 既に変更した箇所を読み返してその意図を思い出したり、さらに変更し直したり

している工数です。

XDDPの開発アプローチには「3点セット」があることでレビューなどによって思い込みや勘違いが相当数除去されますので、テストで発見されるバグも1/5~1/10に減ります。上のケースでは30件程度に減ると予想されます。さらに変更箇所の記録が残っていますので、バグの平均対応時間も半分に減り、手戻り工数は80~100時間程度で済むはずです。

こうして、従来方法で6000時間を要していた工数が、XDDPでは3100時間程度で対応できるということになります。実際にこれを裏付けるデータが多数上がっています。

#### プロフィール

清水 吉男(しみず よしお)

株式会社システムクリエイツ 代表取締役

派生開発推進協議会 代表

ソフトウェアの開発プロセスの改善を中心に活動しています。その中で要求の仕様化と派生開発の効果的アプローチの方法について、それぞれ「USDM」「XDDP」という方法を提案し、これを中心にして普及活動を展開中です。特に「派生開発」を広義に捉えることで、「XDDP」はいろいろな場面に適用できますので、その可能性を広げたいと思っています。

派生開発推進協議会 <http://www.xddp.jp/index.shtml>

## 2. 人材育成

### ■ モダンソフトウェアテストアカデミー開催の背景と狙い

有限会社デバッグ工学研究所  
代表 松尾谷 徹(法政大学兼任講師)

#### 1. 異色のセミナーコース：トップガン教育

日科技連において今までのセミナーには無かった異色のトップガン教育が10月からスタートします。2月8日に行われた講演会では、先進的な企業の社内研修の事例と研修の進め方が紹介されました。ここでは、このコース企画の背景となったソフトウェア業界の人材をめぐる危機的な課題について説明します。

#### 2. ショッキングな調査結果

人材育成の指標となる「ものさし」として、独立行政法人・情報処理推進機構(IPA)のITスキル標準(ITSS)が広く使われています。ご存知のようにITSSでは専門分野ごとに7段階の度合いでレベルを表現します。スキル・レベルを定義するだけでなく、スキル分布の実態調査、すなわち「スキルの見える化」がIPAによって行われています。この調査は平成19年から始まり、21年度の調査の詳細は、5月に発行される「IT人材白書2010～岐路に立つIT人材変革期こそ飛躍のチャンス～」に掲載されております。発行に先だって行われた、ソフトウェアジャパンの講演から作成したスキル分布のグラフを図1示します。

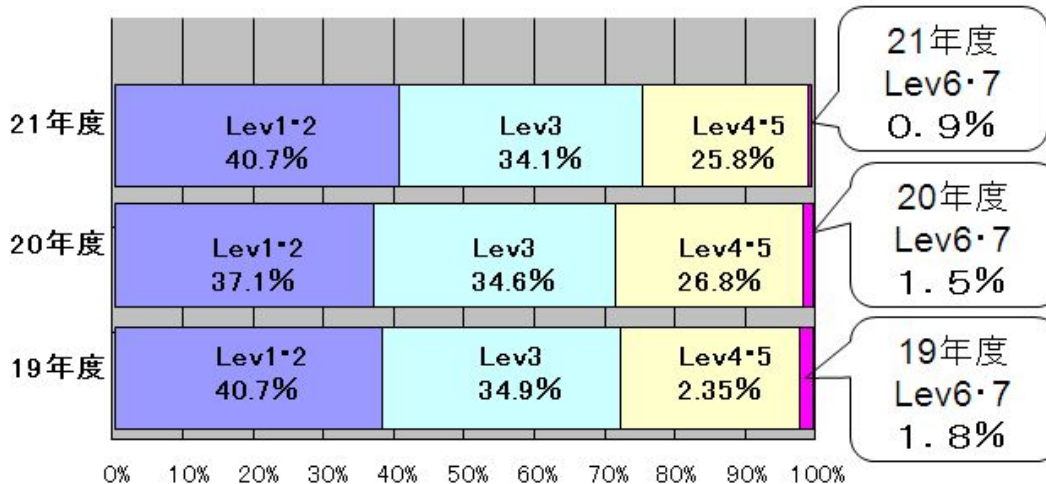


図1. スキル・レベルの分布 IPAの調査より

Lev1・2：実務経験に入っていくための準備期間

Lev3：自立した仕事ができる段階

Lev4・5：特定の経験分野において部下やメンバーに対して技術指導が出来る

Lev6・7：企業内の上下関係が無くとも影響力を持つ真のプロフェッショナル

図1で注目すべきは、高度な技術者であるレベル6・7の占める割合が極端に少なく、たった0.9%しか存在しないのです。しかも、少ないだけでなく、2年間で1.8%から0.9%へと激減し、日本のソフトウェア産業界におけるプロフェッショナルは絶滅危惧種「レッドリスト」に値する状況を示しています。この状況は、産業界や企業にとって危機的であり、絶滅危惧種と同様に回復不能域に入ってしまうと、自力で回復することは絶望的です。

人材の「質」が、製品やサービスの「質」に大きな影響を与えることは明白です。製品やサービスの「質」を見える化することは難しいのですが、その傾向を示すデータが、同じIPAの調査で示されています。それは、ITサービスを購入する企業が、提供側のベンダー企業のIT人材の質を評価した調査です。図2に示すように、「やや不足」と「大いに不足」が90%に達しています。

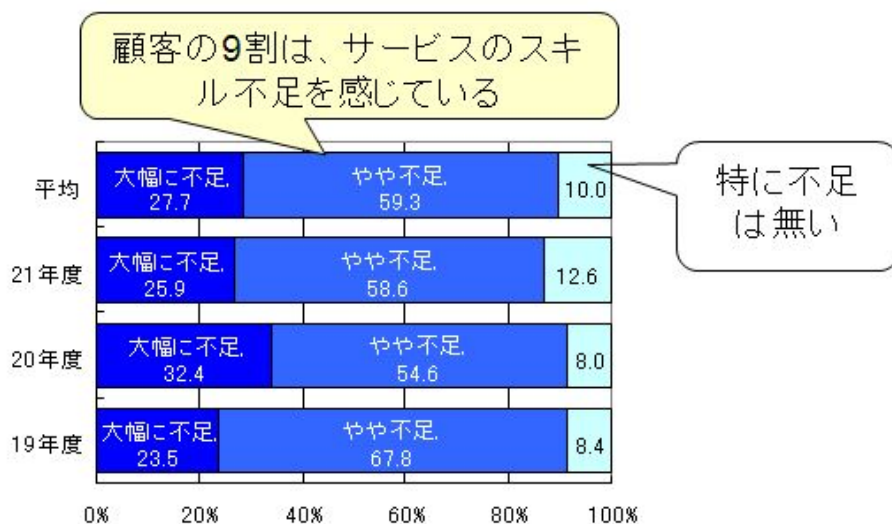


図2. ベンダー企業が評価したIT人材の質

IPA「IT人材白書2010」概要より  
[http://www.ipa.go.jp/jinzai/itss/activity/2010summary\\_of\\_ITHR.pdf](http://www.ipa.go.jp/jinzai/itss/activity/2010summary_of_ITHR.pdf)

### 3. 悲劇的なシナリオ

日本のソフトウェア業界は、ハードウェア(IC やサーバー)や製品ソフトウェア(OS やアプリ)の多くが、グローバルな競争に敗れる中で、かろうじて生き残ってきました。現在まで生き残った原因は、競争が国内に閉じていたからであり、グローバルな競争優位性を持っているわけではありません。海外向けのSIビジネスの多くが苦戦し失敗している現実がこのことを裏付けています。日本の市場は、最近まで供給と需要がバランスしており、質より量を供給することによって拡大を続けてきました。仕事の量が多い一つの原因は、技術進歩とビジネス環境の変化が速く、システムや製品の陳腐を追いかける再構築と移行が続いているからです。

IT サービスの生産モデルは、マンパワーを IT サービスやソフトウェアに変換するものです。マンパワーには、スキルが必要なので頭数だけでなく、質が重要であり、人材の調達と育成が企業の存亡を左右することになります。従来の育成方式は、新人を雇用し、導入教育を行い、OJT と呼ばれる、働きながら育成する伝統的な方法です。少し複雑になりますが、人材育成と価値連鎖をモデル化して図3に示します。

IT サービスの特徴は、単純化すると人的資源をエネルギーとし、ほとんど原料を必要とせずサービスやソフトウェアを提供します。製造業が原料を製品に変換するのと比べ、大きな差があります。製造業の経営感覚で考えると、例えば人的資源を原料と読み替えることによって対比することができます。ただ、原料は長い期間の人材育成による熟成が必要で、熟成段階で技術が変化して陳腐化するリスクも生じます。

図3を使って、企業間の競争について考えてみます。ある企業が提供するサービスの質と量は、その企業が持っている人材のタンクによって決まります。量が不足する場合には、外注することにより社外から調達することも可能です。外注できるリソースは、競争相手の企業も調達できるので、その企業の本質的な競争力にはなり得ません。企業の競争力は、人材のタンクに蓄えられている人材の価値＝人材によって決まります。日本の企業は、終身雇用による長期安定雇用の下で、OJTによる人材の育成(熟成)を行ってきましたが、ITSSのレベル6・7に相当する高度な人材が0.9%しか育っていない事実から、この方式が、進歩の速いIT分野では機能していないことが明らかになったのです。

日本のIT企業では人材の価値連鎖が機能不全に陥っており、量とコストで優れる海外の競合企業は、ベンダーをバイパスし直接顧客へ IT サービスを供給する「中抜き戦略」を展開します。一旦、この流れが生ずると、資金の流れも変わり、流れから孤立した企業は、急激に衰退する悲劇的なシナリオが待っています。



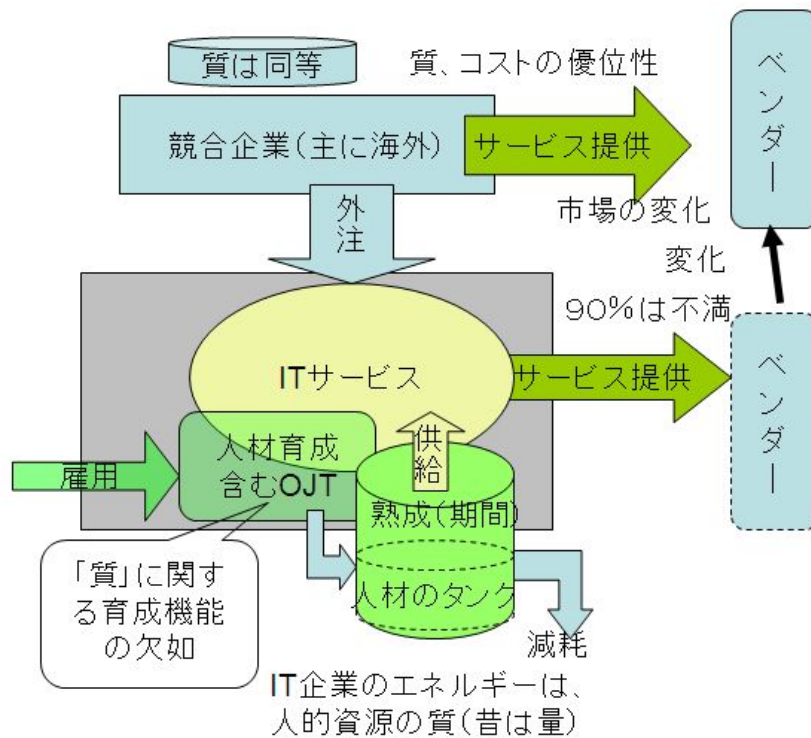


図3. 人材の価値連鎖と競争

#### 4. 人材育成に求められることは

今、何が人材育成に求められているのか、その結論は明らかです。現状のスキル・レベル3程度の人材を大量に育成する底上げ戦略では、図2に示すように、顧客にスキル不足と受け止められており、市場ニーズに合っておりません。さらに、レベル3程度では、コスト的に海外との競争力を急激に失っています。育成への投資を増額して高度人材育成を行うことが困難な経済状況下では、先ず、経営層が戦略的なデシジョンを行い、プロフェッショナル育成を戦略的に位置づける必要があります。

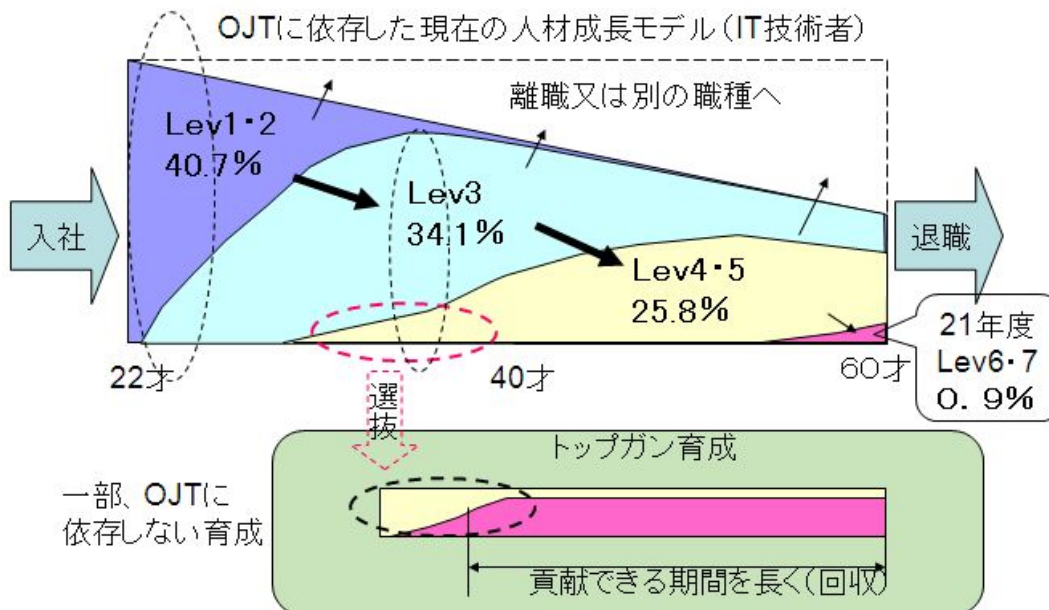


図4. 人材成長モデル: 現在の育成とトップガン育成

図4の上図は、統計的なデータから推測した、平均的な企業の年齢構成とスキル分布を人材育成のトンネルとして示しています。入社からリタイアまでのスキル分布は、成長のスピードと、レベル間の変換率を示しています。課題は、上位レベルへの育成スピードが遅いこと、育成効率が低いことです。次の時代を切り開く、レベル 6・7 に成長する期間があまりにも長すぎ、育成効率(レベルを上げて行く率)が低すぎるのが課題です。

高度人材育成の予算を増やしただけでは、次の時代に企業が生き残るための人材を育成できません。OJT として職務を遂行する中で並行して進んでゆく従来の育成方法では効率が悪いので、合理的な別の育成方法を考える必要があります。

非効率の原因は「仕事を遂行する＝人が育つ」を唯一の手段とする古典的な育成戦略です。よって育成戦略を大きく変える必要があります。何故、レベル 4・5 からレベル 6・7 への進化が遅いのでしょうか。レベル 4・5 に大きな問題が潜んでいます。多くの日本企業は、部下やメンバーへの指導を職位に頼って行う傾向があります。そうすると、高度なスキルを習得するより社内人事評価、すなわち管理的な成果を求める傾向が強くなってきます。

それでは、現代において求められる高度人材育成について考えると、

- ①早く育成し長く貢献する:30代後半にはレベル 6・7 に達成し、実務貢献の期間を長くする。
- ②積極的な投資によるプル型の人材育成:従来の育成方法ではなく、良い指導と良い機会を与え促成する。
- ③ビジネスへの貢献の高い高度人材:市場の変化に追従できる柔軟スキルを持つ。

などが考えられます。

## 5. 解決に向けて

IT 技術者の人材育成は、3段ロケットで表すことができます。1段目は基礎知識と入門的な職場経験で、レベル 1・2 と 3 に相当します。順調に経験を積んで自分の担当範囲の仕事を達成することができます。2段目では、1段目とは違って他者を動かすスキルが求められます。他者に影響を与える範囲は経験と共に拡大しますが、点火するには良いリーダーの下での OJT が必要になります。2段目に点火して順調に成長することがレベル 4・5 に相当します。

最終段である3段目では、職務上の関係を超えた技術的な影響力が求められます。書き物であったり、設計書であったり、コンサルテーションであったり、その媒体は色々ありますが、上司一部下などの階層関係を超えて影響を与えることができるスキルです。2段目で、管理事務や上位の職位に流れ、軌道をそれると3段目に点火することは困難です。現在の IT サービス現場で経験を積むことだけでは、3段目に点火出来ないのが現実です。

昨年度に IPA が行った IT 人材育成強化加速事業報告 ([http://www.meti.go.jp/policy/it\\_policy/jinzai/IT6.htm](http://www.meti.go.jp/policy/it_policy/jinzai/IT6.htm)) の「CDP」の3章におもしろい調査があります。第一線で活躍されている方約90名にインタビューし、技術者のリアルな姿や、成功・失敗段を調査したものです。多くのインタビューでレベル 4・5 の壁を越えるターニングポイントのようなものが語られていることから、3段目が経験年数だけでは点火できない「何か」があることを示唆しています。

ハイレベルな人事育成は、知識教育では意味がありません。自分で考え、問題解決と他者への影響力、めげない意欲などが必要になります。我々は 2005 年からこの種の研修に取り組んできました。研修生が伸びるポイントは、

- ①10年程度の経験を持ち、まだまだやる気のある技術志向の強い技術者を選抜すること
- ②この業界で現役として第一線で活躍しているお手本となる人に接し、考え方や生き方を学ぶこと
- ③自分の考えをまとめ、客観的なレビューを受け、私見ではないロジカルな思考を身につける

などが必要になります。図4の下に示した、トップガンのトンネルを設置することです。

10月から始まるトップガン育成の研修では、なるべく少人数のゼミ形式で、事例など実際の多面的な問題について考え、自分の考えを組み立てる訓練を行います。ゼミの時間とは別に、研修生個別に対応して、課題の整理

から企画書作成まで指導する方法です。2005年から昨年までで3社、のべ200名ほどの研修を行ってきました。その実績から、このトップガン育成のプログラムは、人材をめぐる危機的な課題の一つの解決方法だと考えております。

#### プロフィール

松尾谷 徹(まつおだに とおる)

有限会社デバッグ工学研究所 代表

法政大学 工学部兼任講師

ISO/IEC JTC1 SC7 国内メンバー 同 WG26 幹事

PS(パートナー満足)研究会代表

技術者の知識より、ヒトの側面に興味をもっています。その中でも人材育成、チーム育成などにおける態度変容を促進する教育コンテンツの開発と展開を続けたいと思っています。

## ■ ライフサイクルにおけるプロセス評価と改善から SQuBOK®を読み解く 第4回

株式会社日立システムアンドサービス シニアコンサルタント  
古賀 恵子  
日本ユニシス株式会社 品質保証部  
大川 鉄太郎  
プロムス 代表  
河合 一夫

### 第4回の内容

連載4回目は、「ライフサイクルにおけるプロセス評価と改善」です。ソフトウェアライフサイクルにおけるプロセス評価と改善はどのような目的でどう行われるのか、ソフトウェアの品質を考える上でどのような意味を持つのかなどを、古河課長と熊川さん[[第1回「本連載の進め方」](#)]に語って貰います。

### ソフトウェアプロセスアセスメントとは

熊川さんは、課長に呼ばれて次の調査を命じられました。

「Y取締役が、『わが社のソフトウェアライフサイクルプロセスが、標準に概ね整合していることはよく分かった。しかしプロセスが規定されているだけでは絵に描いた餅だ。日々の仕事のプロセスどおりに行われ、期待される成果を上げているのか、継続的な改善を進めているのか、についても知りたい。』とおっしゃっている。熊川君、社内のプロセス評価と改善の現状を至急調査して改修すべき点があれば報告してくれ。」

熊川さんは、どう手をつけてよいか分からなかったので古河課長に聞きました。「課長、この仕事にどう取り組んだらよいか分かりません。私の今の知識と経験では手に負えそうにないのですが…。」古河課長は「一人で抱え込んだまま時間を無為に過ごしてしまわなくなったのは良いことだ。これは大きなテーマなので、少し長くなるが詳しく説明するからよく聞いてくれ。」と言って説明を始めました。

「ソフトウェアプロセスを評価する目的は、現状を体系的に把握して改善を進めることにある。そして結果として製品品質や開発効率を向上させるためだ。そのきっかけはざっくりいって2つある。一つは、実際に発生した品質問題を引き起こした根本原因を究明し、再発防止策を立案する過程でソフトウェアプロセスを評価する。もう一つは、ベストプラクティスの体系に照らしてプロセスを評価して、足りない部分を改善していく場合だ。この両方をバランスよく実施していくことが必要だ。」古河課長は本論に入っていきます。

「プロセス評価については、SQuBOK®の「2.3 プロセスアセスメント・プロセス改善のマネジメント」の記述と参考文献を参照すると全体のイメージがつかめる。また体系的なプロセス評価については、歴史的な背景を少し知っておくとよいだろう。(コラム参照)

代表的なプロセス評価モデルであるCMMIとISO/IEC 15504には大きな違いがある。CMMIはソフトウェアのベストプラクティスで構成されるアセスメントモデルであり、業種固有の要件などを取り込むカスタマイズは認めていない。15504はそれ自体がアセスメントモデルではなく、アセスメントモデルとアセスメントを規定するメタモデルだ。その要件を満たせば自由に固有の要件を取り込んでいくことが可能なため、業界や業種ごとのアセスメントモデルが次々に開発されてきている。」

熊川さんはすぐに疑問が浮かんできました。「CMMIやISO/IEC 15504と、先日伺った『ISO/IEC 12207(JIS X 0160)ソフトウェアライフサイクルプロセス』や、これを下敷きにして日本独自に作成したモデルである『共通フレーム2007』(共に連載第2回を参照)とは、どのような関係になるのですか?」古河課長はうれしそうに答えました。「良い質問だ。ただ結論から言うと直接の関係はない。CMMIは独自に定義している固有ゴールと共通ゴールを達成しているかどうかに関心があるからだが、結果としてかなりの部分をそれらのプロセスモデルはカバーしている。ISO/IEC 15504はメタモデルだが、実はそこで想定されている標準的なプロセス参照モデルは、ISO/IEC 12207やISO/IEC 15288だ。だからこれまで作られたアセスメントモデル(コラム参照)の多くはISO/IEC 12207を下敷きにして、そこに業種固有の要件を反映している。」

最後に古河課長はこう締めくくりました。「『プロセス評価と改善』というテーマへのアプローチは選択肢が多い。君はまずそれらの選択肢を評価して、結果を報告したらどうだろうか？アプローチによっては、すぐに億単位の予算が必要になるからだ。」

熊川さんは「全体が何とか見えてきました。何か特に注意すべき事項はありますか？」と聞きました。古河課長は「『プロセス評価と改善』というテーマへ取り組む際の重要な留意事項が2つある。1つ目はよく言われる形骸化だ。これは会社の大きな無駄につながるので、なぜそうしたことが起きるのかを自分でよく考えながら進めてくれ。ポイントは、経営層から現場までが一貫して「プロセス改善を本気で進める」という意識を共有し続けることができるかどうかだ。現場が「上位層の本音は、手をかけずに形を整えることだ」と考えると、上位層が必要な予算を出し渋ると、言い訳材料ができて簡単に形骸化してしまう。2つ目は、改善は継続的であることだ。我々の仕事の外部環境(市場、顧客、競合…)も内部環境(組織体制、要員、能力、方針…)も常に変化しているので、仮にある時点で最適化を達成できたとしても、次の瞬間から最適ではなくなっているからだ。」と答えました。熊川さんはこの2つのポイントを肝に銘じながら調査に着手しました。

#### <プロセス評価の流れ…>

1980年代に日本では問題点を起点とするボトムアップ型のQC活動による改善活動が中心だった。1990年代以降はベストプラクティスの体系、言い換えるとあるべき姿のモデルに照らしてプロセスをアセスメントするアプローチが注目され、日本だけでなく世界中に普及して来た。

この中でアセスメントの基盤とするモデル(=アセスメントモデル)が数多く提案され、使用されてきた。日本のソフトウェア業界ではISO9000シリーズが最初に注目されたが、全業種対象のため抽象度が高いことから、90年代に盛んにソフトウェアに効果的に適用する方法について議論された。一方ソフトウェアとシステムを対象とするSW-CMM、CMMIが、経済産業省の「日本版CMM構想」の影響もあって、次第に注目されるようになった。この大きな流れは各国で余り差はないが、具体的な姿は地域によってかなり違う。

アメリカは、国防総省(DOD)の影響もあり、一貫してSW-CMM、CMMIが中心になってきた。ヨーロッパでは、90年前後から数多くのモデルが使用されてきたが、現在では15504系のモデルであるAutomotive SPICE(自動車産業)、SPICE for SPACE(航空宇宙産業)、Medi SPICE(医療機器産業)、Banking SPICE(金融業界)、などがCMMIと共に展開されている。この中で例えば自動車業界では、欧州の主要完成車メーカー(Audi, BMW, Daimler, Fiat, Ford, Jaguar, LandRover, Porsche, VW, Volvoなど)が、車載ECU(Electronic Control Unit)に向けたソフトウェアを開発するサプライヤーに対して、開発プロセス標準として定義したAutomotive SPICEに基づいたアセスメントを実施している。日本の主要自動車部品メーカーもこれらの完成車メーカーに部品を輸出しているため、このアセスメントを受けている。

「ライフサイクルにおけるプロセス評価と改善からSQuBOK®を読み解く」シリーズは今回で終わりです。SQuBOK®はソフトウェア開発のいろいろな場面で活用できることを知っていただけたでしょうか。SQuBOK®には先人たちのさまざまな知恵がいっぱい詰まっています。ぜひ一度手にとってみてください。

#### プロフィール

古賀恵子(こが けいこ)

株式会社日立システムズアンドサービス シニアコンサルタント  
ソフトウェア開発に関する品質管理支援を行っている。

大川鉄太郎(おおかわ てつたろう)

日本ユニシス株式会社 品質保証部  
ISO/IEC SC7 WG10 委員、情報技術標準化研究センター(INSTAC) SPA WG 委員  
サービス品質の測定とマネジメント、ソフトウェアプロセスの改善、信頼性に取り組んでいる。

河合一夫(かわい かずお)

プロムス 代表  
PMI 日本支部地域サービス委員会副委員長、  
ソフトウェア技術者ネットワーク(S-open)副会長  
ソフトウェア開発プロジェクトの支援やソフトウェア開発支援ツールの開発を行っている。

## 4. トピックス

### ■ (1)2009 年度(第 25 年度)ソフトウェア品質管理研究会・分科会成果発表会の報告

日科技連 SQiP 事務局

執筆協力: 池田浩明(インテック株式会社)

前回のトピックスで、ソフトウェア品質管理研究会(通称:SQiP 研究会)の 2009 年度における一年間の研究会活動を振り返り、さらには 2010 年度の活動予定を紹介しました。今回は 2009 年度の活動の締めくくりである最終例会(2/26)で実施した分科会成果発表会のご紹介をいたします。

成果発表会では、一年間活動してきた分科会がそれぞれ発表を行い、研究会主査、副主査がその研究内容を評価した上で投票し、最優秀賞ならびに優秀賞2分科会を決定します。

今年度は、以下の3つの分科会が最優秀賞ならびに優秀賞を受賞しました。

研究成果論文は、以下に記す日科技連HPに掲載していますので、是非ご覧ください。

### ■2009年度研究会テーマ「今こそ、現場と学問に根ざしたソフトウェア品質の追求」に沿って進めた一年間の研究の中で最も高い評価を受けたのは、第5分科会テスト設計グループ！

#### 最優秀賞

第5分科会 「テスト設計グループ」

研究テーマ:「派生開発における影響箇所の把握改善によるテスト範囲の特定方法の提案」

研究成果:[http://www.juse.or.jp/software/167/attachs/5.1\\_report.pdf](http://www.juse.or.jp/software/167/attachs/5.1_report.pdf)

池田浩明研究会副委員長からのコメント:

- 派生開発をさらによりよいものにするために、機能間マトリクスという手法を新たに考案した点は高く評価できます。
- 機能間マトリクスによる影響箇所の特定手法には新規性が感じられます。
- 比較実験等により、提案手法の有用性を評価することを期待したいと思います。

#### 優秀賞

第2分科会

研究テーマ:「プロジェクト描写のための情報構造～正確なプロジェクト把握のために～」

研究成果:[http://www.juse.or.jp/software/167/attachs/2\\_report.pdf](http://www.juse.or.jp/software/167/attachs/2_report.pdf)

池田浩明研究会副委員長からのコメント:

- 進捗報告の不正確性を情報構造の観点から分析するという試みはユニークです。
- 丁寧でわかりやすい内容であり、組織の現状分析や改善のための道具として実際の現場で活用できる可能性を感じます。
- 進捗報告以外のプロジェクト情報も関連づけるとよい実際的になるでしょう。

#### 優秀賞

第3分科会

研究テーマ:「設計者自身による設計品質作り込みーテストエンジニア視点の活かし方ー」

研究成果:[http://www.juse.or.jp/software/167/attachs/3-2\\_report.pdf](http://www.juse.or.jp/software/167/attachs/3-2_report.pdf)

池田浩明研究会副委員長からのコメント:

- 非常に面白い研究で、実証までしている点には感心しました。
- さらに一歩踏み込んで、テスト容易性の高い設計のあり方などへの発展を期待したいと思います。

#### プロフィール

池田 浩明(いけだ ひろあき)

株式会社インテック 技術本部 技術部

ソフトウェア品質管理研究会運営小委員会委員

ソフトウェア品質シンポジウム委員会委員

プロセスの標準化と改善、開発環境整備、プロジェクトの品質改善支援などを行う。

## ■ (2)ソフトウェア品質シンポジウム 2010(SQiP2010)参加募集のご案内

～ソフトウェア品質に関する国内最大のイベント～

ソフトウェア品質シンポジウム委員会  
委員長 小笠原秀人

ソフトウェア品質シンポジウムは、ソフトウェア開発に関わるさまざまな方々が一堂に会し、現場で役立つ実践的な技術や経験、ノウハウ、研究成果を発表し意見交換を行う場です。今年は、SQiP設立30周年を記念するイベントとして開催いたします。

併設チュートリアルでは、ソフトウェアの品質向上に大きく寄与する技術を取り上げました。基本を学び、開発現場での品質向上技術を習得する場として活用してください。また、企画セッションとクロージングパネルでは、SQiPの活動にこれまで大きく貢献されてきた方々に参加していただきます。現在を知り、過去を振り返り、未来へ向かうための気持ちを高める場としてぜひ活用してください。そのほか、基調講演、特別講演、チュートリアル、企画セッション、招待講演、SIG(Special Interest Group)によるグループ討論、ツール展示など、バラエティに富んだ内容を企画しました。

経験論文、経験発表では、今まで以上にソフトウェア開発の現場で実践しているさまざまな種類の発表を提供いたします。SQiP(スキップ)シンポジウムが、実践事例を発表する場として定着しつつあることに感謝しております。

ソフトウェアの高品質化に対する要求が、今後ますます増してくるのは確実です。地に足のついた実践的な品質向上活動を積み重ねることと並行して、今までのやり方を大きく変えることへのチャレンジも求められていると思います。

ぜひ、このシンポジウムで次につながる何かをつかみ取ってください。そして、自分たちの活動にフィードバックして、その結果をまたこのシンポジウムで発表していただければと思います。ソフトウェア品質シンポジウムが、ソフトウェア品質に関するさまざまな問題を解決するための“最初の一歩”を提供できれば、このうえない喜びです。

**日 程:** 2010年8月25日(水)～27日(金)

**会 場:** 東洋大学・白山キャンパス(東京・文京区)

**参加費:** 39,900円(一般)／37,800円(日科技連賛助会員)

※早期申込を実施します!!(期間:開始～7月中旬まで)

※8月25日(水)の併設チュートリアルは別料金となります。

**後 援:** 文部科学省、独立行政法人情報処理推進機構(IPA)

**プログラム:**

基調講演: 根本 武彦 氏 (株式会社三菱東京UFJ銀行 常務取締役)

特別講演: 高橋 伸夫 氏 (東京大学大学院 教授)

その他、一般発表、企画セッション、チュートリアル、SIGなどを予定しております。

申込先／問合わせ先:

財団法人日本科学技術連盟 教育推進部 第二課

ソフトウェア品質シンポジウム2010担当

TEL.03-5378-9813 FAX.03-5378-9842 E-mail: sqip-sympo@juse.or.jp

▼詳細はWebサイトをご参照ください。

<http://www.juse.or.jp/software/160/#01>

## ソフトウェア品質シンポジウム2010委員会

### 委員長

小笠原 秀人 (株式会社東芝)

### 副委員長

堀 明広 (株式会社NTTデータMSE)

森崎 修司 (奈良先端科学技術大学院大学)

渡辺 晴美 (東海大学)

### 委員

青木 利晃 (北陸先端科学技術大学院大学)

足立 久美 (株式会社デンソー)

藤井 彩乃 (株式会社インテック)

脇谷 直子 (広島修道大学)

佐藤 慎一 (株式会社NTTデータ)

小井土 亨 (株式会社OSK)

河野 哲也 (電気通信大学大学院)

佐々木 方規 (株式会社ベリサーブ)

服部 京子 (日本アイ・ビー・エム株式会社)

中田 雅弘 (株式会社日立製作所)

田 哲 (テクマトリックス株式会社)

野中 誠 (東洋大学)

服部 彰宏 (富士ゼロックス株式会社)

三條 香 (富士通株式会社)

永田 敦 (ソニー株式会社)

佐藤 光紀 (株式会社日本オープンシステムズ)

## プロフィール

小笠原 秀人(おがさわら ひでと)

株式会社東芝 ソフトウェア技術センター プロセス・品質技術開発担当 主査

ソフトウェア品質シンポジウム委員会 委員長

SQiP ステアリング委員会委員

SQuBOK V2 策定部会 委員

ソフトウェア生産技術(メトリクス活用、不具合管理、静的解析、テスト設計/管理、プロセス改善など)に関する研究・開発およびそれらの技術の推進・展開活動を実践している。



### ■ (3)「第5回世界ソフトウェア品質会議(5WCSQ)」論文募集まもなく開始します！



日科技連 SQiP 事務局

WCSQ (World Congress for Software Quality) は、全世界のソフトウェア品質専門家が集まって幅広く議論を行う国際会議です。ASQ (The American Society for Quality)、EOQ (The European Organization for Quality)、JUSE (財団法人日本科学技術連盟)の3団体がWCSQを運営しています。

第1回会議を1995年にアメリカで開催したのを皮切りに、これまで、第2回(2000年)を横浜、第3回(2005年)をドイツ・ミュンヘン、第4回(2008年)をアメリカにて開催しました。第5回目となる2011年11月の会議は、中国・上海にて開催いたします。

著名な経営者や、第一線で活躍するエキスパートらによる基調講演のほか、ソフトウェア品質について議論が交わされるパネルディスカッションなど、盛りだくさんの内容を予定しています。

論文発表セッションでは、一般論文の発表に加え、各国のシンポジウムなどで優秀な評価を得た「ベスト・オブ・ザ・ベスト」論文の発表も企画しております。また、本会議の開催前日には、様々なチュートリアルも予定しております。

3年に一度、世界のソフトウェア品質専門家が一堂に会するWCSQに、ぜひ論文を投稿してみませんか！

## 第5回世界ソフトウェア品質会議(5WCSQ)

会 期 : 2011年10月31日-11月4日

開催地 : 中国・上海 ナレッジ イノベーション コミュニティ (KIC)

<http://www.kic.net.cn/>

テーマ : 齊心協力 追求卓越 - United Under One Banner: Best of Best Quality

ひとつの旗の下に団結しようー卓越した最高品質を目指してー

### 論文募集テーマ

以下のトピックに関する論文を募集しますが、これら以外でも、ソフトウェア品質に関わる論文を広く募集します。

- (1) ソフトウェア品質の基本:ソフトウェア品質モデル、品質保証、品質改善、品質特性(ディペンダビリティ、ユーザビリティ、セキュリティ、セーフティ)、V&V
- (2) 組織レベルのソフトウェア品質マネジメント:プロセスと活動、品質マネジメント組織、プロセスアセスメント、プロセス改善、品質アセスメント、監査、教育・育成
- (3) プロジェクトレベルのソフトウェア品質マネジメント:調達マネジメント、構成管理、リスクマネジメント、プロジェクトマネジメント、品質計画
- (4) ソフトウェア品質技術:メトリクス、レビュー、テスト、ベストプラクティス
- (5) ソフトウェア品質分析と評価:信頼性予測、障害分析、品質データ分析
- (6) 運用・保守:ITIL、SLMとSLA、インシデントマネジメント

## 募集論文

募集する論文は、以下の2つのタイプです。

- 一般論文：企業などでの実践経験に基づく研究、ケーススタディ、理論研究などをまとめた論文です。分量はA4用紙で最大8ページまでです。
- ベスト・オブ・ザ・ベスト論文：5WCSQの論文募集テーマに関連する内容で、国内の関連シンポジウムなどで優秀な論文として表彰された実績のある論文について、推薦を受け付けます。

論文提出方法、提出スケジュールなどは6月に公開予定です。その他詳細な情報は決まり次第、随時更新いたします。

5WCSQの案内はこちら：<http://www.juse.or.jp/software/154/>

## 5. 憩いの広場「体力を科学する」

### ■ 第6回 その場でできる心身評価

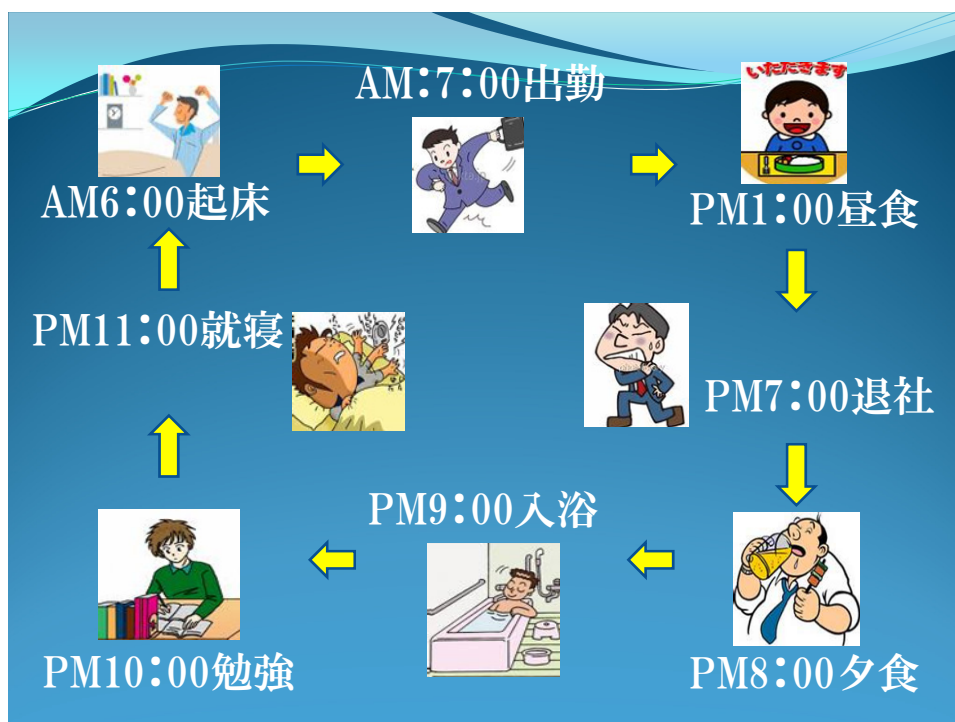
#### ～「ストレス」「筋力」「柔軟性」「体のサビ」に対する対処法②～

<監修>

清泉クリニック整形外科 スポーツ医学センター  
専務理事 脇元 幸一

パソコンに向かってデスクワークしているみなさん、肩こり・腰痛・頭痛などに悩まされていませんか？第5回では「ストレス」「筋力」「柔軟性」「体のサビ」に対する対処法①として「**食う**」＝「**食事**」についてご案内しました。第6回目となる今回は、「ストレス」「筋力」「柔軟性」「体のサビ」への対処法②として「**寝る**」＝「**睡眠**」について紹介いたします。

みなさんの生活は全て何らかのサイクルによって成り立っていて、人間が健康な生活を送るためには、このサイクルを乱すことなく生きていくことが大前提となります。では、このサイクルとは何を示すのでしょうか。サイクル＝「リズム」、つまりサイクルとは生活のリズムを表します。例として私の一日を記載してみます。



みなさんも毎日の生活を思い返してみてください。個人差はあれど、みなさんの生活はリズムを刻みながら成り立っているものです。記載例は一日のリズムですが、大きなリズムでは「1週間」「1ヶ月」「1年」「10年」、小さなリズムでは「12時間」「1時間」「1分」「1呼吸」など、必ずなんらかのリズムで行っているのです。

日々の全てを同じリズムに整えることが最良の手段となりますが、なかなかそうはいかないこの世の中…。

そこでリズムを刻む上で最も調節が可能なのが**睡眠**なのです。

睡眠のリズムについて勉強してみましょう。睡眠には大きく分けて二つのサイクルが存在します。みなさんも聞いたことがあるかもしれません。

- ・ Non-REM 睡眠(ノンレム睡眠)

「体」は起きているが、「脳」が休んでいる状態であり、俗に言う“深い眠り”です。

この時には成長ホルモンが分泌されます。成長ホルモンは子供の筋肉や骨の成長もそうですが、大人でも胃腸や肌の修復を促すのに重要であり、新陳代謝、免疫機能、ストレス除去にも関与しています。

・REM睡眠(レム睡眠)

「脳」は起きているが、「体」が休んでいる状態であり、俗に言う“浅い眠り”のことです。

この時の脳は覚醒(起きている)時と変わらない状態であり、身体の回復をはかっている時間となります。



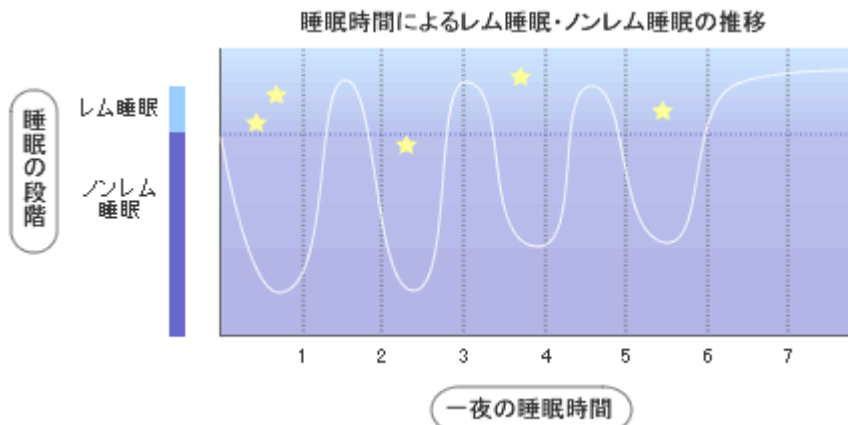
このREM睡眠・Non-REM睡眠の繰り返しは睡眠であり、脳と体の休息リズムなのです。みなさんも必ず経験していると思いますが、寝起きが爽快のときや、起こされても不快では無いという経験があると思います。逆に、すぐ起きられないときや、起こされるとすごく不快に感じる時があったはず。そうです！合点のいく皆さんなら想像ができたと思いますが、寝起きが良い時は“浅い眠り＝体の休息”の時であり、寝起が悪い時は“深い眠り＝脳の休息”の時となります。

目覚めのタイミングとしてNon-REM睡眠では休んでいた脳が無理やり起こされるので、頭がぼーっとし、すっきりとしません。逆にREM睡眠の段階では脳も活動しており眠気が浅いので、目覚めのタイミングに適しています。ですから、起きるときは“浅い眠り＝体の休息”時に起きることでストレスなく起きることができるのです。

睡眠の種類	休息	眠り
Non-REM睡眠	脳の休息	深い
<b>REM睡眠</b>	<b>体の休息</b>	<b>浅い</b>

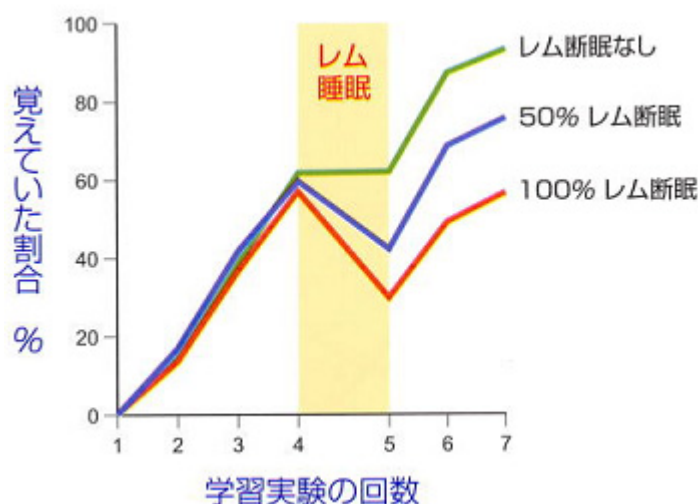
では、この睡眠リズムの日常生活における実践方法をご案内します。

寝始めはNon-REM睡眠へ向かい、1.5時間後にREM睡眠のピークを迎えます。その後Non-REM睡眠へ移行し1.5時間後にまたREM睡眠のピークを迎えるという1サイクル1.5時間＝90分のリズムを繰り返しています。つまり、起きる時間を“浅い眠り＝体の休息”のピークに合わせ、逆算して布団に入るといふようにすると良い目覚めが迎えられます。



下の図に示すように、REM睡眠がしっかり取れると学習効果が向上することが証明されています。言い換え

ば、仕事の効率をあげるためには REM 睡眠をしっかりと取ることが重要であり、睡眠がみなさんの昇進にも関係するのでは！？



一般的な書物では 7 時間の睡眠を取ることが望ましいと言われています。睡眠のサイクルを示すグラフを参照していただくとわかりやすいと思いますが、寝始めてから 7 時間後はちょうど REM 睡眠のピークとなるのです。しかし、仕事で忙しくて 7 時間の睡眠が取れない方は、起床のタイミングを REM 睡眠のピークに合わせるというちょっとした工夫で心身へのストレス軽減が図れます。

REM 睡眠のピークで自然に目が覚めるというのが本来の自律神経リズムにストレスをかけない唯一の方法です。目覚まし時計で起きるような生活を止め、寝る時間と起きる時間をしっかり一日のリズムに組み込むことが最大の健康を享受できるともいえます。

今後は目覚まし時計の無い生活を目標としましょう！！

第 6 回となる今回は、「ストレス」「筋力」「柔軟性」「体のサビ」に対する対処法②として「寝る」＝「睡眠」についてご案内してきました。次回は、「ストレス」「筋力」「柔軟性」「体のサビ」対処法③として「遊ぶ」＝「運動」について紹介いたしますので楽しみに！！

(文責：清泉クリニック整形外科 スポーツ医学センター 理学診療部 嵩下 敏文)

#### プロフィール

嵩下 敏文(だけした としふみ)

清泉クリニック整形外科 理学診療部 部長

NPO 法人 FINE 理事

「慢性疼痛疾患の治療」は現代西洋医学にできないこととして位置づけられており、対処療法による医療が現状となっています。慢性疼痛疾患は我々医療従事者が治療するものではなく、痛みを有する者自身が治療を行わなければならない疾患であり、その慢性疼痛疾患の根本治療体系の確立を目指し日々の臨床、および 研究活動を行っております。

本誌の全部、あるいは一部を無断で複写複製(コピー)ならびに転載することは、著作権法上の例外を除き、禁じられています。