

## Software Process

"Should-be" vs "As-is"

Kouichi Kishida  
SRA-KTL  
@ SPIN Workshop in 浜松  
Dec 11-13, 2008

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

3

## Prescript

- 今回は新しいSlideを用意するつもりだったのですが、SEA-SPIN 発足以来10数年間にいろいろなミーティングで使った Slide を眺めているうちに、それらを編集しなおすのが面倒になったので、簡単なプレゼンのシナリオを考えて、適当に古い Slide を間に挿入するという形にしました。

- 以下、当日使った Slide + アルファ です。

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

2

## Dijkstra のコメント

たしか60年代半ば

- ー テスティング(デバッグ)はバグの存在を証明することはできるが、決してその不在証明にはなりえない
  - ・ あきらかに Should-be
  - ・ プロセス品質 → プロダクト品質

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

3

## Postscript

Dijkstra さんの発言の真意は:

- ー Program の実行プロセスの表現(すなわちプログラムの書き方は、それを読む人が正しさを検証できるものでなければならないということ。つまり、プロダクトとしてプログラムの品質は、プロセスの表現形式によって定まるということだったので、それがかれの構造化プログラミングの主旨。

わたしの構造化プログラム設計は、それとは少し違って、プログラム実行プロセスの表現をなるべく単純なパターンに簡約化することを狙っていました。たぶんジャクソンやワーニエの構造化手法も同じだったと思います。品質(プログラムの正しさやパフォーマンス)についての意識は、当時のわたしには、あまりありませんでした。保守性についてはかなり強く意識していましたが、

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

4

## Royceの論文

70年夏 @ Wescon

- ーハードウェア・システム開発のWaterfall Model を紹介
- ーそれをソフトウェアに適用することの難しさを論じた
  - ・ しかし Should-be としての Waterfall を推奨したものと誤解された
  - ・ プロダクト品質重視(?)

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

5

## Postscript

1970年8月に初めてアメリカへ行ったとき、いきなり飛びこんだ Wecon の併設セミナーで Royce さんの講演を聴きました。そのころはプロセスについての関心はあまりなかったので、「なるほどそんなものか」と思っただけでした。

その後、Royce さんの真意とはかけはなれて Waterfall というコトバだけが一人歩きを始めた。

1987年の 9th ICSE @ Monterey で、Bob Balzer とわたしが Most Influential Past Paper の表彰を始めたとき、この Paper を第1号に選んだのはそういった意味合いでした。9th ICSE の Proceedingsに再録されているので、ぜひ多くの人に読んでもらいたいものです。

昨年4月の SEA Forum では、落水先生に解説をお願いしました。

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

6

## Lehmanの進化論 70年代後半

- S-type & E-type という分類
- ソフトウェア進化論の諸法則
- 開発プロセスから進化のプロセスへ
  - わたしが初めて聞いたのは78年秋
  - 変化(進化)する As-is Process

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

7

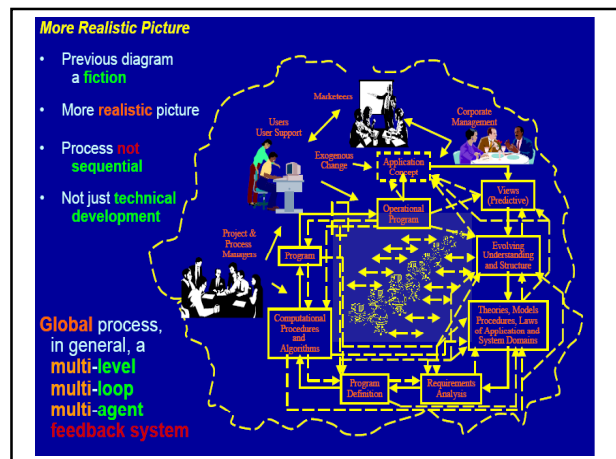
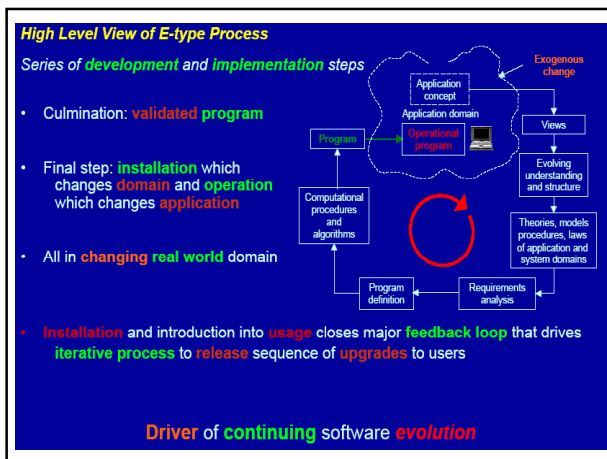
## Postscript

Lehman 先生とはその後長いお付き合いになりました。  
4th ICSE で session chair をやらされたり、ISPW のコミュニティに ひっぱりこまれたり、...  
Software Evolution Dynamics から教えられたもっとも重要なポイントとは、プロセスの動的な側面を重視すること、ソフトウェア・システムは無限のパラメータをもつ現実世界から有限の仕様を切り出して作ったもの似すぎないのだということを常に忘れないように、ということでした。  
先生の Web Page は: <http://www.doc.ic.ac.uk/~mml/>  
そこにはたくさんの論文や Presentation Slide が載っています。

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

8



## Postscript

- Lehman 先生のこの Slide は、ソフトウェア・プロセスを考えるさいに、Should-be ではなく As-is を重視すべきことをはっきりと示していると思います。
- Multi-level, Multi-loop, Multi-agent の Feedback System であるというのが As-is Software Process の特性。
- それは Should-be というかたちでは表現できない。

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

11

## Jackson & Macracken SEN April '82

- Life Cycle Concept Considered Harmful
  - 過去の遺物であり、それにもとづくマネジメントの強制はよくない。
  - エンドユーザとのコミュニケーションを阻害する
  - 技術者・管理者の自由な発想を妨げる
- プロセスのダイナミックな変化を重視したコメント

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

12

## Postscript

1980年に開かれたConference & Workshop (テーマは「これからのソフトウェア開発を考える」)に参加した後のPostscriptとして書かれたもの。このエッセイ全文は、ACM Portalに入っています:

<http://portal.acm.org/citation.cfm?id=1005937.1005943&coll=GUIDE&dl=GUIDE&CFID=14529832&CFTOKEN=14323090>

ソフトウェア開発問題を考えるにさいして Life Cycle という概念がもたらす危険性についての2人の警告は傾聴するに値すると考えられるのですが、しかし、いまだに少数意見にとどまっているようですね。

しかし、1908年初頭に、やがて到来するであろうエンド・ユーザ主導のシステム開発の問題を予見したあたり、さすがだと思います。

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

13

## ISPW

1980年代半ばから後半にかけて、ソフトウェア・プロセスに関する議論が盛んに行われました。その中心は1984年にLehman先生の呼びかけではじめられた ISPW (International Workshop on Software Process)でした。

84年の第1回(London)は、いろいろな人がいろいろな意見をとりどりに述べて面白かった。BoehmさんのSpiral Modelの話も聞いたのもこのWSでした。また、いまテストや品質保証に関して引用されているV字型モデルの話も、このWSでポーランドのTurski先生とLehman先生が提案したのが最初だったと思います。その後しばらくして日本のだれかが、まるで自分が発案したかのようにV字型モデルの話をするのを聞いて唖然とした記憶があります。

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

14

## 2nd ISPW

85年にLos Angeles 郊外のリゾート施設で開かれた第2回WSではいくつかの印象的な出来事がありました。

1つは、プロセス・モデルを最上位に置き、それのもとづく開発方法論、さらにはそれを支援するツールという階層構造を議論の枠組みとしたというプログラム委員会の意向に対して長友Bob Balzerが真っ向からObjectionをとなえたことです。「わたしはこれまでは違う新しいツールを開発したいと考えている。もしそうしたツールが実現されれば、プロセスはこれまでとまったくちがうものになるはず。だからツールの地位をそのように貶めてはいけません」。このObjectionでWSの議論全体が大きく変わってしまった!

もう1つはMITだかの若い研究者が、あまり面白くないプレゼンをしたのですが、参加者のほとんど全員がそのプレゼンの途中でどんどん部屋を出ていってしまったこと。WSの厳しさを感じました。わたし自身はFlower Modelのアイデアを発表しようと考えていたのですが、PC Memberから「それはあまりにContext Freeすぎる。やめとけ」といわれて断念しました。

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

15

## 3rd ISPW

第3回のISPWは、1986年の秋に、翌年の9th ICSEのPC Meetingと併設するかたちで、ロッキーの山の上(海拔4000メートル近く)のスキー・リゾートで開催された。やたら呼吸が苦しかったことを覚えている。

このWSでは、その後のソフトウェア工学の行方に影響を与えた2つの重要なプレゼンテーションが行われた。

1つは、当時まだIBMにいたWatts HumphreyさんのCMMのアイデア(まだCMMという名前ではなく、ソフトウェア開発におけるプロセス・マネジメントのフレームワークの提言)。そしてもう1つはLee Osterweilさんのプロセス・プログラミングのアイデアだった。

どちらかをICSEのKeynoteにしよう、Balzerさんと話し合って、結局Osterweilさんにしよということに決めたのでした。

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

16

## Humphrey's CMM

1986秋 @ 3rd ISPW

- Material Labor Model
  - ・ Waterfall Type の Life Cycle Model にもとづいている
- PSP → TSP → CMM という枠組みは儒教(朱子学)のパラダイム
- もともとIBM社内の大規模プロジェクトでの経験をまとめたもの。それがDoDの発注業務を支援するために応用された。

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

17

## Postscript

何人かのIBM関係者の方々の話をきくと、Humphreyさんは「生まれつきのマネージャだ」とのこと。

そうした管理者の視点からすれば、CMMのアイデアはまさしく適切なものだといえるでしょう。世界最大のコンピュータ・ユーザ(ソフトウェア発注者)であるDoDが、組織成熟度という枠組みを気に入ったのも当然。

しかし、かつてISPW in 函館やSPIN WS in 静岡で指摘したように、その概念的枠組みが儒教(朱子学)のマネジメント・パラダイムとあまりによく似ている点が、わたしには気に入らない。

P

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

18

## 儒教とソフトウェア工学

10年ほど前の ISPW in 函館 での討論のなかで、「おや、これは、むかし読んだ儒教哲学の古典に書いてあったことと似ているな」と気づいた。しかしながら、「論語」や「大学」は、いまの日本では、もう死書ののひとつに数えられていて、ソフトウェア技術者のなかでそれらを読んだことのある人はほとんどいないだろう。だから PSP や TSP が新鮮に感じられるのではないだろうか？

その後、SPIN WS in 静岡で、そのことを指摘したのだが、みなさんの反応はいまひとつだったように思われる。

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

19

## 大学における マネジメント・パラダイム

「大学」に書かれた8段階パラダイムは次の通り:

古之欲明明徳於天下者、先治其国。

欲治其国者、先齋其家。

欲齋其家者、先修其身。

欲修其身者、先正其心。

欲正其心者、先誠其意。

欲誠其意者、先致其知。

致知在格物。

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

20

## 奇妙な対応関係

- 平天下: ISO15504
- 治国: CMM
- 齋家: Team Software Process
- 修身: Personal Software Process
- 正心: 正しい方向づけ
- 誠意: まじめに
- 致知: ソフト工学の知識 (\*\*BOK) を学ぶ
- 格物: オブジェクト指向 !?

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

21

## Beyond 儒教

儒教的マネジメントの問題点は、その論理構造がきわめて美しくできていて、多くの人びとを魅了すること、しかし、現実に応用するとすぐに腐敗すること、である。

そのことは、このパラダイムによる政治が行われてきた中国・朝鮮そして日本の歴史を振り返ってみれば明らか。

その限界を乗り越えるヒントは、18世紀日本の社会哲学者・荻生徂徠やその批判的継承者・富永仲基によって示されているように思われる:

徂徠: 「弁名」 & 「弁道」

仲基: 「出定後語」 & 「翁の文」

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

22

## Process Programming

やはり1986秋 @ 3rd ISPW

- ICSE'87 の Keynote
  - with Lehman's Counter Talk
- プロセスのなかの固定された手順はたしかにプログラム可能、しかし .....
- 「プロセスもソフトウェアだ」というなら、ソフトウェアの本質的な属性である変化を どうとらえるのか？

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

23

## Postscript

Process Programming のアイデアは、たしかに魅力的であり、90年代に多くのソフトウェア工学研究者を虜にした。

その結果、いくつかのいわゆる Process-centered Development Environmentが試作されたが、そのほとんどは旧来のエンジニアリングパラダイムにもとづくものであり、あらかじめ規定された手順による開発を支援するものでしかなかった。

9th ICSE の Proceedings に収められた Osterweil さんの基調講演とそれに対する Lehman さんの反論を対比しながら読むとおもしろい。

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

24

## Floyd の論文

SEN April, '88

- Product 指向から Process 指向へのパラダイム・シフト
- ソフトウェア・システムの開発・運用・保守のプロセス全体を、システム進化の視点でとらえることの必要性を指摘した。

<http://portal.acm.org/citation.cfm?id=43846.43851&coll=portal&dl=ACM&CFID=15115520&CFTOKEN=24391439>

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

25

## Postscript

1986年の秋に SIGMA プロジェクトの運営方針をめぐる MITI と衝突し、プロジェクトから足を洗って暇になったので、何人かの人たちに声をかけ、寄付金を募って、産学・国際共同の研究プロジェクト SDA を立ち上げて、数年間、ソフトウェア設計プロセスを支援する環境について、連続ワークショップ形式の討論を行った。

その途中で目にしたのが、この Floyd 女史のエッセイ。

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

26

## Postscript

Floyd さんは、「プログラムとは?」、「システムとは?」、「品質とは?」、「ソフトウェア開発とは?」、「エラーやユーザの能力とは?」、「プログラムの理解やドキュメントの役割とは?」、「開発方法論とは?」といった項目ごとに Product 指向と Process 指向という2つの視点の違いを対比している。

たとえば「品質」についていえば、前者の視点ではそれはプロダクトの属性であり、プログラムからユーザを見るかたちで定義されるが、後者の視点では、プロダクトを利用するユーザ・プロセスの属性であり、ユーザの目でプログラムを眺めるかたちで定義される。

では、「プロセスとは?」という問いに対しての2つの視点の違いは?

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

27

## Motivation

ソフトウェア開発は Immaterial Labor そして Affective Labor の典型的な例である。であるとすれば、その仕事に携わる技術者のモチベーションがきわめて重要な役割を果たす。

しかし、モチベーションは定量化や把握がむずかしい。したがって、多くの場合、定量化が容易なほかの要因の影に隠れてしまっている。ソフトウェア・プロセスに関する問題を考えるには、それを実行するエイジェント(技術者)のモチベーションを正しくとらえる必要がある。その意味でイギリスの研究者たちの行った MoMSE Project はきわめて興味深い。

<http://homepages.feis.herts.ac.uk/~ssrg/MoMSE.htm>

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

28

## McConnell's Comment

Motivation is a soft factor:

It is difficult to quantify, and it often takes a back seat to other factors that might be less important but are easier to measure.

Every organization knows that motivation is important, but only a few organizations do anything about it. Many common management practices are penny-wise and pound-foolish, trading huge losses in motivation and morale for minor methodology improvements or dubious budget savings.

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

29

## MoMSE

Modeling Motivation in Software Engineering  
Sponsored by EPSRC, UK (2005)

Goal: To develop a fundamental model for software engineers motivation

Method: Survey of related papers using SLR

SLR: Systematic Literature Review  
もともと医療分野で開発された文献踏査の技法

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

30

## MoMSEの調査結果 (Motivators)

- Challenge
- Change
- Benefit
- Problem solving
- Team work
- Science
- Experiment
- Development practice

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

31

## Postscript

ソフトウェアの本質的な特性は「耐えざる変化」である。

開発途上で発生するさまざまな変化(仕様や設計の変化, 開発環境や技法の変化, プロジェクト組織の変化, etc)は, 通常ネガティブな意味合いでとらえられているが, この調査結果が示すところによれば, そうした変化は, しろ技術者の挑戦意欲を高めるものとしてポジティブにとらえるべきものではないだろうか。

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

32

## MoMSEの調査結果 (De-Motivators)

- Software process
- Life Cycle
- Maintenance

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

33

## Postscript

ソフトウェア工学を構成する基本的な概念は「プロセス」および「構造」の2つである。  
前者は時間を含む動的な概念であり, 一方, 後者は時間を含まない静的な概念である。

これまでのソフトウェア工学の歴史をふりかえてみると, 各種の構造化技法, さらにそれに続くオブジェクト指向などの方法論は, いずれもプロダクトの構成要素(モジュール, オブジェクト etc)への切り分けとその相互関係すなわち構造を重視するものであった。いいかえれば「時間からの逃走」というべき流れであった。

ウォーターフォール以降に提案されたさまざまなプロセス・モデルも, 時間の持つダイナミックな特性を極力殺して, 起こりうる変化やそれを克服するための挑戦的な要因を消してしまっているあたりが, 技術者のモチベーションを阻害している結果になっているのであろう。

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

34

## SPI(プロセス改善)

- SPI は, 現在のプロセスが理想の Should-be Process と比較して「よくない」という価値判断から出発する。
- 理想の実現を目指して前進しようということである。
- 残念ながら, 現実には決して理想に到達することはできない。
- その結果, SPI は抑圧的な永久革命運動という負のサイクルに陥ってしまう。

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

35

## As-is Process

- 現状否定的な価値判断をやめて, もっとポジティブに考えよう。
- いまのプロセスは決して悪いものではない。
- 別に改善の必要はない。
- もちろん, 解決すべきさまざまな問題がある。それらはたえず変化し続ける周囲の環境条件に由来する。
- 問題解決は改善とは別物である。

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

36

## As-is の進化

- 現在のプロセスは、プロジェクトを取り巻くさまざまな社会・経済・技術的環境に対して、われわれができるかぎりの最適化を行った結果である。
- もしそこに何か問題があるとすれば、それはプロジェクトを取り巻く環境に起因するものである。
- そして、環境はたえず変化し続ける
- すなわち As-is はすぐに As-was に変わってしまう。

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

37

## 課題

- ソフトウェア・プロセスのダイナミックスをどのようにとらえたらよいか？
- その結果としてできあがるプロセス・モデルの私たちは、いったいどんなものになるであろうか？
- わたし個人は、かつて Flower Model および Traveling Model という試案をいくつかのワークショップで提案したことがあるが、いずれもまだ「叩き台」の域を出ていない。

Dec 11-13, 2008

(C) Kouichi Kishida @ SPIN Workshop in 浜松

38