SEAMAIL

Newsletter from Software Engineers Association

Vol. 13, Number 4 April, 2002

目　次

編集部から		1
ソフトウェアおよびソフトウェア開発の構造改革	筏井 美枝子	2
遠隔学習の概況	黒澤 公雅	19
ICU2001年冬期ソフトウェア開発論授業総括	新谷 勝利	33
論文募集 - ISFST2002		38

ソフトウェア技術者協会
Software Engineers Asociation

ソフトウェア技術者協会 (SEA) は，ソフトウェアハウス，コンピュータメーカ，計算センタ，エンドユーザ，大学，研究所など，それぞれ異なった環境に置かれているソフトウェア技術者または研究者が，そうした社会組織の壁を越えて，各自の経験や技術を自由に交流しあうための「場」として，1985年 12月に設立されました．

その主な活動は，機関誌 SEAMAIL の発行，支部および研究分科会の運営，セミナー／ワークショップ／シンポジウムなどのイベントの開催，および内外の関係諸団体との交流です．発足当初約 200人にすぎなかった会員数もその後増加し，現在，北は北海道から南は沖縄まで，500 余名を越えるメンバーを擁するにいたりました．法人賛助会員も 25社を数えます．支部は，東京以外に，関西，横浜，名古屋，九州，広島，東北の各地区で設立されており，その他の地域でも設立準備をしています．分科会は，東京，関西，名古屋で，それぞれいくつかが活動しており，その他の支部でも，月例会やフォーラムが定期的に開催されています．

「現在のソフトウェア界における最大の課題は，技術移転の促進である」といわれています．これまでわが国には，そのための適切な社会的メカニズムが欠けていたように思われます．SEA は，そうした欠落を補うべく，これからますます活発な活動を展開して行きたいと考えています．いままで日本にはなかったこの新しいプロフェッショナル・ソサイエティの発展のために，ぜひとも，あなたのお力を貸してください．

代表幹事： 深瀬弘恭

常任幹事： 荒木啓二郎 高橋光裕 田中一夫 玉井哲雄 中野秀男

幹事： 伊藤昌夫 大場充 落水浩一郎 窪田芳夫 熊谷章 小林修 桜井麻里
酒匂寛 塩谷和範 篠崎直二郎 新谷勝利 新森昭宏 杉田義明 武田淳男
中来田秀樹 野中哲 野村行憲 野呂昌満 端山毅 平尾一浩
藤野誠治 松原友夫 山崎利治 和田喜久男

事務局長： 岸田孝一

会計監事： 橋本勝 吉村成弘

分科会世話人 環境分科会(SIGENV)：塩谷和範 田中慎一郎 渡邊雄一
教育分科会(SIGEDU)：君島浩 篠崎直二郎 杉田義明 中園順三
ネットワーク分科会(SIGNET)：人見膺 松本理恵
プロセス分科会 (SEA-SPIN))：伊藤昌夫 塩谷和範 高橋光裕 田中一夫 端山毅 藤野誠治
フォーマルメソッド分科会(SIGFM)：荒木啓二郎 伊藤昌夫 熊谷章 佐原伸 張漢明 山崎利治

支部世話人 関西支部：臼井義美 小林修 中野秀男 横山博司
横浜支部：野中哲 藤野晃延 北條正顕
名古屋支部：筏井美枝子 石川 雅彦 角谷裕司 野呂昌満
九州支部：杉田義明 武田淳男 平尾一浩
広島支部：大場充 佐藤康臣 谷純一郎
東北支部：布川博士 野村行憲 和田勇

賛助会員会社：ジェーエムエーシステムズ 東芝ＩＴソリューション ＳＲＡ ＰＦＵ
テプコシステムズ 構造計画研究所 富士通 オムロンソフトウェア
キヤノン 富士通エフ・アイ・ピー 新日鉄ソリューションズ
ダイキン工業 オムロン 富士電機 ブラザー工業 オリンパス光学工業
リコー アルテミスインターナショナル NTTデータ ヤマハ
福井コンピュータ 日本ネスト オープンテクノロジーズ ＳＲＡ西日本
日本総合研究所
（以上25社）

ＳＥＡＭＡＩＬ Ｖｏｌ．１３，Ｎｏ．４ ２００２年4月8日発行 編集人 岸田 孝一
発行人 ソフトウェア技術者協会（ＳＥＡ）
〒１６０-０００４ 東京都新宿区四谷３－１２ 丸正ビル５Ｆ
T: 03-3356-1077 F: 03-3356-1072 sea@sea.or.jp
印刷所 有限会社 錦正社 〒１３０-００１３ 東京都墨田区錦糸町４－３－１４
定価 ５００円 （禁無断転載）

編集部から

☆

今度はやや早めの船便です．

☆☆

昨年の暮れと今年の１月に SEA 名古屋支部で行われた議論のあらましを，筏井美枝子さんにまとめていただきました．筏井さん，そして関係者のみなさん，ありがとうございました．

☆☆☆

今年入会されたばかりの黒澤公雅さんからは，教育分科会で報告された遠隔学習の現状についてのサーベイをまとめていただきました．なかなか有益な情報だと思います．

☆☆☆☆

幹事会メンバーの１人である新谷勝利さんからは，ご自分が ICU で教えていらっしゃる「ソフトウェア開発論」講座についての生のデータを提供していただきました．これからの教育問題を考える上で貴重な情報でしょう．

☆☆☆☆☆

例年秋の恒例行事である ISFST in 中国の論文募集要項が決まりましたので，巻末に載せました．今年は，当初古都・西　安で開催の予定だったのですが，楊子江ダムが年末に完成し，名勝・三峡が水没して見られなくなってしまうという話で，急拠計画を変更し，プレ会議チュートリアルを西安で，そして本会議を武漢でという「二都開催」に変わりました．みなさん，ふるって論文/事例報告/ツールデモをご応募ください．

☆☆☆☆☆

SEA名古屋支部 討論会　レポート

ーソフトウェアおよび、ソフトウェア開発の構造改革　ー

筏井　美枝子

SEA名古屋支部では、1995年4月から、毎月1、2回の割合で、ミニセミナーと呼ばれる勉強会を開いています。(これまでの、ミニセミナーの背景や内容につきましては、*http://sea.forums.gr.jp/nagoya/*を参照)

夕方18:30－20:30の2時間に開催している勉強会（セミナー、フォーラムなど）には、いつも15名～25名くらいの参加者があり、コーヒー&お菓子つきで、和やかな中にも活発な意見交換が行われています。勉強会の後は講師を囲んで懇親会となりますが、話がはずむと、何次会かに名古屋名物の味噌煮込みうどんを食べて、お開きになるのは午前1時2時ということも珍しくありません。

本音で語る勉強会であるのと、スタッフが時間的に余裕のないこともあって、レポートのたぐいはほとんど残していません。今回苦戦していますこのレポートも、本来は出す予定ではなかったのですが、昨年12月26日のフォーラムに参加されていた岸田さんから、「SEAMAILにレポートを」と言われたことをうけまして、パネリストの方々にご相談しましたところ、諸岡隆司さんと田中一夫さん、それに東京から参加してくださった熊谷さんの三人が書いてくださるということでしたので、なんとかレポートにまとめてみることにしました。

今年1月22日の討論会と合わせて、三人のプレゼン資料と感想、そして、私のおぼつかない記憶をたどりまして討論の経過をお伝えしたいと思います。なお、この2回のイベントは、忘年会・新年会を兼ね、with Beer & Pizaで行われましたので、私を含めアルコールに弱い何人かは、舌が勝手に回ってしまったり、一部の記憶が脱落しておりますので、つじつまが合わない箇所がありましても、みなさまの豊かな想像力で行間を埋めてくださいますよう、お願いいたします。

☆ SEA名古屋支部12月ミニセミナー ☆

ー　年忘れフォーラム　ー

日時:　2001年12月26日（水）18:30 - 20:30

タイトル:「ソフトウェアおよび、
ソフトウェア開発の構造改革」

概要:　近年、ソフトウェアの開発は、

- 期間の短縮
- 仕様変更の対応
- 品質の確保
- コストの削減

といった要望が、ますます厳しくなってきています。

今回のミニセミナーは、そのような要望に「どのように対応していくのか？」、「対応可能なのか？」という疑問や問題点を、XP, リファクタリングなど、今話題の開発方法を交えて討論します。なお、今回は年末恒例の年忘れフォーラムということで、パネラの方に火付け役になっていただき、参加全員で寒さを吹き飛ばす熱い議論を展開したいと思っています。

☆ 2002年1月ミニセミナー ☆

ー　新春 討論　ー

日時:　2002年1月22日（火）18:30 - 20:30

タイトル:「ＸＰはＨＹＰＥか？
－ これからのソフトウェア開発環境・プロセスのありかたをさぐる －

概要:　１２月のフォーラム「「ソフトウェアおよび、ソフトウェア開発の構造改革」では、ＸＰの可能性が話題になりました。ネットニュースグループでも、ＸＰの導入可否をめぐって、賛否両論、今様々な話題が沸騰しています。そこで、今回は、前回のフォーラムの続編として、XPとは何か？そして、いま、ＸＰにまつわりどんなことが議論されているかなどを簡単に紹介し、それをきっかけにこれからのソフトウェア開発環境・プロセスのありかたをさぐっていきたいと思います。

ところで、「構造改革」って何？

－　二つのフォーラムを振り返って　－

筏井　美枝子

小泉さんが「構造改革」という言葉を使ってから、「‥の構造改革」という言葉が、一時期、そこここで聞かれ、この企画を立てたころは「構造改革ばやり」だった。最近は、疑惑議員の話題ばかりが目に付く。それはそれで、腐敗しきったたものが次々に明るみにでて、これをきっかけに政治の構造改革がどっと進むかという期待も少しはあったのだが、ちかごろは腐敗をたんまり抱えていそうな構造同士の防衛戦じみてきて、改革への期待がはぐらかされなきゃ良いがと善良な国民の一人として心配になる。

とここまで書きながら、うかつにも、小泉さんが「構造改革」という言葉をどこから持ってきたか知らないことに気づいた。英語のrestructureだろうか？いわゆるリストラだけど、リストラは従業員の首切りみたいなイメージが定着してしまったから、本来の意味を新しい言葉で置き換えたのかもしれない。ついでに、改革という言葉をWEBの辞書で引いたら、「基盤は維持しつつ、社会制度や機構・組織などをあらため変えること」とあった。いずれにしろ、問題は、「改めて、どう変えるか？」なのだが、それにはあるべき構造の姿がなくてはならない。少なくとも、改めるべき方針を明確にする必要がある。でないと、改革の果てに、すっきりした入れ物にはなったが、その構造を支えるはずの人間には牢獄のようなものになってしまう可能性だってある。

無駄と思えるものを片っ端から省き（それが本当に無駄なのかはさておき）、組織としての枠と上下階層を固め、責任範囲を明確にし、生産性向上を目指す。それも確かに構造改革といえるのかもしれない（というか、一般にはこの種のものが主流であるのだろう）が、何か息苦しい。「貧すれば鈍す」ともいえる雪印に始まる食品業界のあきれ返ったモラルのなさも、実は食品業界に限った話しではないという声もある。不始末が起きれば、明確になっているはずの責任範囲はどこへやらで、工場長や現場の人間に責任をなすりつける経営陣の破廉恥さ。組織とか構造とはいったい何なのだろうか？

前置きが横道にそれて長くなってしまったが、昨年１１月に入ったころ、名古屋支部恒例の年末フォーラムを何にしようかという話しが出たとき、KIUIの中で、リファクタリングとかＸＰに興味を持っている人がいて、これを材料にしようということになった。でもこれだけのフォーラムだとつまらないよねということになり、忘年会も兼ねているしというので、「ソフトウェアおよび、ソフトウェア開発の構造改革」などと、かなり大仰で胡散臭いタイトルにした。

なぜ胡散臭いかといえば、政治の構造改革もそうだけど、構造の実体や改革の方針が明確でない限り、この言葉は単なるお題目でしかない。しかし、逆にフォーラムのタイトルとしてみれば、どんな構造に対してどんな改革が飛び出すか、蓋をあける楽しみも出てくるというものである。

さてその結果はといえば、議論の詳細は次ページを読んでいただくとして、パネリストの田中さんから産業構造の改革という面からの鋭い提案をしていただいたにもかかわらず、いわゆる構造改革的な話題は盛り上がらず、主な話題はＸＰなど、イメージが見えるもののほうに集中した。

というわけで、１月のフォーラムはＸＰの話題を中心にしたのだけれど、ＸＰを代表とするAgileな方法論が出現した背景を見ていったり、オープンソース潮流に始まる一連の流れを考えるなかで、意外にも、ある種の草の根的な構造改革が垣間見えてきた気がする。

改革のキーは次の二つである。

・時代や環境にマッチした柔軟な構造に
・人間に重点をおいた構造に

である。

「構造」とか「組織」という言葉は、何故か無機質めいて硬く冷ややかである。たとえば、鉄筋建でガラス張りの建造物は、堅固で効率的で美しくあれば良いというかもしれない。しかし、人間の営みに関わる構造という視点でみた場合は変わるのではないだろうか。それらが何のためのもので、何を支えているかを考えたとき、世の中の構造、特に組織にあたるものはもっともっと柔軟でなければならず、かつ、中心には「人間」という有機質の暖かさを据えなくてはならないように思う。そして、それをしっかり考えた改革こそが真の構造改革なのだという気がする。

‥‥

開催から二ヶ月が過ぎ、「構造改革」というテーマに照らして、今回の二つのフォーラムをすこしだけ振り返ってみました。読み返せば、胡散臭いものをさらに胡散臭くしてしまった気もしますが、忘年会＆新年会フォーラムのレポートということで、大目にみていただければと思います。

経過をかいつまんで

昨年12月26日のフォーラムは、パネリストのかたから、次のようなプレゼンテーションがあった。

諸岡隆司@中電コンピュータサービスさん:

ＸＰやリファクタリングなど最近注目されている技法/方法論について説明し、ソフトウェア自身の構造改革案を提示。

田中一夫@jfitsさん:

ソフトウェア開発の管理面（プロセスを含む）での構造改革案を提示。

海老原祐太郎@りぬくす工房さん:

最近の開発経験から、ソフトウェアならびにプロセス面での、構造改革案を提示。

諸岡さんと田中さんの資料ならびに感想は、後のページに掲載した。海老原さんには、当日、WEBシステム開発における問題点を中心にお話しいただいた。

討論では、ＸＰに関する意見・質問が多くあがった。以下に思いつくものを列挙する。

・現状の開発には使えない

あれはSmalltalkだからできるのだという意見と、事務処理システムは本当は仕様変更はそれほど多くないから必要ではないという意見があった。

・部分的には使えるのでは？

ユニットテスト、ペアプログラミングなどが、使えそうな候補としてあがった。

・見積もりはどうなるのか？

開発の始めにすでに予算と納期ありきの開発が多く、開発の途中途中で細かく調整していくようなやりかたは現実には無理との声が多かった。

・現在の開発プロセスでリファクタリングの機会は作れるのか？

一旦作られて動いているシステムにリファクタリングのために手を入れるのはユーザに嫌がられるうえに、コストをかけて行う必要性を上に認めてもらうのは難しいため、現状では、正式なリファクタリングの機会は持ちにくい。結局、品質向上への関心により、技術者個人が機会をひねり出すしかないだろうというのが大方の意見だった。

・誰もがソースコードを修正できるようにするのは危険ではないか？

誰がいつソースのどこを修正したかは、構成管理をしっかりしていればわかる。リリース版の責任者は必要だろうというのが、オープンソースでソース管理をしている人の意見であった。

・設計はどうなるのか？

XPでは、従来の設計やドキュメントにあたるものがないが、ある程度は必要ではないかという意見が多かった。

・ペアプログラミングは本当にうまく行くのか？

実際にやってみた人から、相手が誰かにより、うまくいく場合と苦痛を極める場合があるという、非常にもっともな意見があった。

また、構造改革に関しては、「ソフトウェア開発の構造改革は、素養のない技術者を全部辞めさせることである。」などというブッ飛んだ意見から、まずは、きちんと構成管理を徹底すべきという地道な意見（参加者の中で、現在関わっている開発で構成管理をしていないという人の数が以外にも多かった）までさまざまであったが、「構造改革」の構造にあたる対称と構造の定義があいまいであったのと、司会者の力不足もあり、議論は、ソフトウェア開発技法、CMMを含む開発プロセス、業界の構造など、多岐に発散した形で終わった。

このフォーラムで、ＸＰへの関心が多かったこともあり、今年の1月22日には、ＸＰに焦点を合わせ、事例やネットニュースでの話題を探るなどして、もう一度討論会を行うことにした。とっかかりとして、

・ＸＰの位置づけ

Martin Fawlerの"The New Methodology"というWEB上の記事をもとに、新しいソフトウェア開発方法論（*Lightweight Methodology*）の流れと、その中でのＸＰの位置づけを説明。

・ＸＰのプロセス流れ

ＸＰに関するWEBサイトはいくつかあるが、http://www.extremeprograming.orgから、XPのプロセス図に従い、どんなふうに開発が行われるのかをたどってみた。

・ＸＰの導入事例など

*"November/December, 2001, IEE SOFTWARE"*にＸＰの導入事例が掲載されていたので、その中から事例を三つ紹介。

を筏井が説明することになった。その際、質問・討論を適宜行いながら進めた。

結局、ＸＰは、ある条件（小規模、仕様変更が多い、顧客が協力的、技術者が前向きなど）の基では、効果的な方法論であるらしいことがわかったが、一方で、この条件にあてはまるというのが、現状では結構難しいこともわかった。（特に、顧客が協力的、技術者が前向きという点で）また、繰り返し（iteration）の中で、リファクタリングとCRCカードとペアプログラミングにより設計しか行われないという点も、不安や疑問を覚える人が多かったように思う。

参加者の多くは、「うーむ、今の現状ではすぐには使えそうにないな。」という感触を持ったようにみえる。若い技術者の人たちも、現状とのギャップに戸惑い気味な感じで「是非やってみたい」という人は、少なかった（個人的には、ちょっと残念..）。

ＸＰを含むAgile（機敏な）開発方法論が、これからのソフトウェア開発に構造改革を起こすものかどうかは、今のとことまだわからない。ただ、世界の複雑さ、変化の速さ、そしてそこに住む人間の価値観に合わせて、技術や方法論も変化していくのは自然なことであり、機敏さ、人間性、多様さといったキーワードが構造改革のキーかもしれない..。

感想と資料

田中　一夫@ Jfits

SEA 名古屋支部 12 月ミニセミナーに参加して

ひょんな事から、12 月ミニセミナーを申込みました。すると、筏井さんから以下の Mail がきました。そして、驚く事に、パネラーになっていました。

パネラのかたには、一人１５分づつのプレゼンをお願いしています。
テーマはですね。すごいですよ。腰を抜かさないでね。
「ソフトウェアおよび、ソフトウェア開発の構造改革」
です。構造改革ばやりなので、わざと胡散臭さを匂わせたりしましたが。^^;
で、田中さんには、ソフトウェア開発の管理面（プロセスを含む）でのうんと刺激的な改革案を出してもらえるれば良いです。田中さんなら簡単でしょ。:)

追い込まれてしまったので、慌てて、次頁の PPT を作成しました。この中では、

- **コスト削減～高い費用の Vender**
- **ソフト開発の期間短縮**
- **品質の確保**
- **仕様変更の対応**

の４要件に関し、簡単に書きました。

当日のパネル後の議論でも多少触れられましたが、大きな課題は日本のソフトウェア産業の歪みであると考えます。

今回、客先でのプロキシー顧客の位置付けで、客先常駐となり、2000 年 4 月から 2001 年 3 月まで某プロジェクトに参画しました（このプロジェクトのために、名古屋単身生活が始まった）。Vender 側の体制は、１次受けから４次受けまであり（実際には、５次受けまであったらしい）、1 次受け企業は、モノ作りというより、管理面に徹していたように見える。この１次受け企業は、世間では SI Vender として大手であり、旧来の建設業で言うところのゼネコンである。

では、具体的に今回の体制で何が問題になったかを整理してみると、まず「コスト高」という問題が大きくクローズアップされる。今回の Vender は、初期構築費用はある意味で妥当であったが、保守・維持費用は高いと云わざるを得ない価格であった。3 次受けあるいは 4 次受けの会社の技術者が出した見積に比べ、2 倍？の価格で提示されているようだ。これには、管理費や初期構築の Lost 回収分等が入っていると思われるが、ユーザ側としては、これ以上の保守・維持を継続すべきかどうか判断に悩む金額になってしまった。

次に、問題になったのは「品質の悪さ」であり、開発時の Bug の多さは凄かった。この原因は、寄せ集め集団であり、コミュニケーションの悪さと技術・知識のバラツキから品質が悪くなったと思われる。さらに、保守・維持環境になっても相変わらず初歩的な Bug が発生していた。品質の悪さの全てが、下請け構造だとは思えないが、下請け構造だと許される仕組みになっていると思われる。

さらに、この様な Vender を採用したのが、SI Vender 大手だと安心できるというユーザ側の認識の甘さからきている。特に、このユーザは、この SI Vender に過去、何回か依頼して「納期は必ず守る」という実績があったらしい。

「作る側も良くないが、発注する側も良くない」というのが最近の傾向だと思う。XP(eXtreme Programming) が流行っているのは、従来、一方的に作る側が悪いって思考だったが、XP ではユーザ負担・責任を明確に謳っており、作る側は、作る事に徹することができそうだから、作る側の採用が増えているのではないだろうか。

今回、筏井さんのお陰で自分の位置付けを再考させられ、少しだけ、今後の方向が整理できたのは幸いであった。

ソフトウェアおよびソフトウェア開発の構造改革

2001年12月26日

自己紹介

・コンピュータ業界で20数年
・MainframeのProgrammingから開始

・某山一證券のシステム部門
・倒産後、日本フィッツに入社

・1999年より某トヨタ自動車に派遣
・TS3 カードのプロジェクトに参画
・主に、証券・保険のモール構築
・ユーザサイドの立場

・現在は、某トヨタFS証券のプロキシー客

と云うことで、今回はユーザ側での発言

今回の筏井さんからの指示

パネラのかたには、一人15分づつのプレゼンをお願いしています。

テーマはですね。すごいですよ。腰を抜かさないでね。

「ソフトウェアおよび、ソフトウェア開発の構造改革」

です。構造改革ばやりなので、わざと胡散臭さを匂わせたりしましたが。^^;

で、田中さんには、ソフトウェア開発の管理面(プロセスを含む)でのうんと刺激的な改革案を出してもらえるれば良いです。田中さんなら簡単でしょ。:)

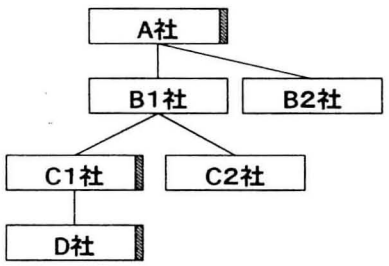
Copyright © 2001 by Kazuo Tanaka All Rights Reserved.

コスト削減～高い費用のVender

2001年12月26日

改革案

1. 開発Venderの階層構造



＜発注側＞
・顔が見えない会社と管理だけの会社は除外すべき
(ユーザ側の体制強化が必要)

＜受注側＞
・下請けではなく、対等な業者間取引

・総勢、10数名　・要件打合せ参加は ▌ の5名
・A社は2名　・B1社の顔は見えない

ソフト開発の期間短縮

2001年12月26日

改革案

2. XP は事務システムで使えるか

・XPの特長として、
　・レス ドキュメント
　・ユーザ責任
　・テスト ファースト
　等があり、賛同するが
・XPの適用分野として
　・要件の曖昧なモノ
　・Compilerの不要な言語
　があり、事務システムへの全面適用は無理？

・全面適用は無理でも、
　・レスドキュメント
　・ユーザ責任
　などは、採用すべき

Copyright © 2001 by Kazuo Tanaka All Rights Reserved.

品質の確保

2001年12月26日

改革案

3. プロセス改善

・ソフトを作る立場として、メトリクス軽視は許せない
・プロセスは、開発企業で確立すべき
・Programmer個人の技術・知識を上げる責任は個人にある
・自浄能力の無い会社は、無くなる

・当たり前の事を、必ず行う会社を選ぶ
・CMMに拘ることなく、プロセス改善を行う会社を選ぶ

Copyright © 2001 by Kazuo Tanaka All Rights Reserved.

仕様変更の対応

2001年12月26日

改革案

4. 変更は常にある

・ユーザ側に居ると、システム仕様は、直前まで固まらない
・紙ベースでの要件詰めは、限度ある
・XPで推奨している「シンプルデザイン～できるだけ明日への設計をしない～」は、疑問

・ユーザ責任は、増やすべき
・全面適用は無理でも、
  ・レスドキュメント
  などは、採用すべき

Copyright © 2001 by Kazuo Tanaka All Rights Reserved.

ソフトウェアおよびソフトウェア開発の構造改革

2001年12月26日

まとめ

筏井さんの期待する、改革案では無いが

・ユーザ(受注側)の技術・知識の向上
・ユーザに適切な人が居ない場合は、Soft Houseから履う

・Soft Houseの営業に惑わされない

Soft House側

・中間搾取を行うような仕事はしない
・自浄努力
・プロセス改善は、メトリクス採取を行えば、やるでしょう。。。

Copyright © 2001 by Kazuo Tanaka All Rights Reserved.

諸岡隆司@中電コンピューターサービス

【セミナーに参加して一言】

XP .... どうなんでしょうね。

XPは、ソフトウェアを作る上で、とてもいいポイントを重視した開発方法だと感じ、私が携わっているプロジェクトでも、試行的に実施しています。

しかし、今回のセミナーに参加して、XPを行うにはいろいろな問題点があることを指摘され、とても参考になりました。

契約や納品の問題、ユーザ側の体制など、なかなか現状の開発環境では難しいかもしれません。

ただ、ソフトウェアを作っていて、

「実働しているシステムだから、触らない」

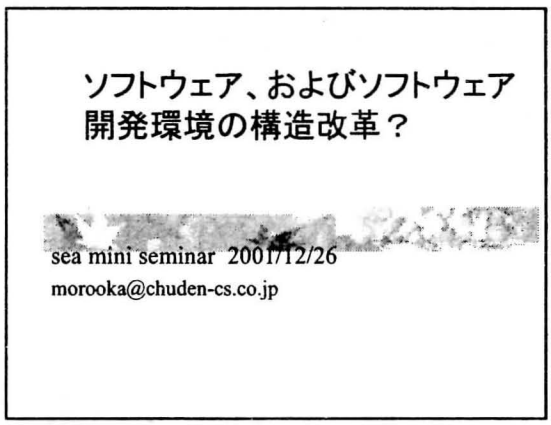
「仕様変更は別途、期間と費用を見積もらさせて頂きます」

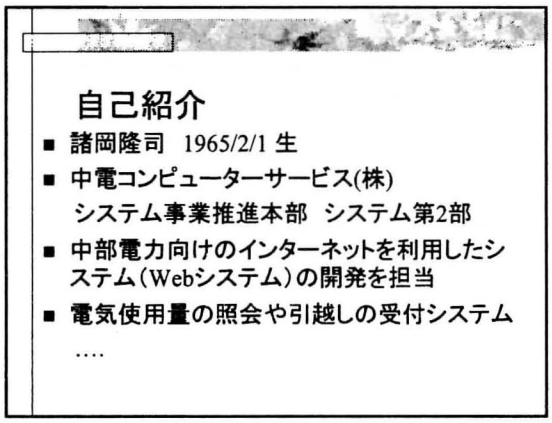
「変更は少しですが、テストにかなり時間を要します」

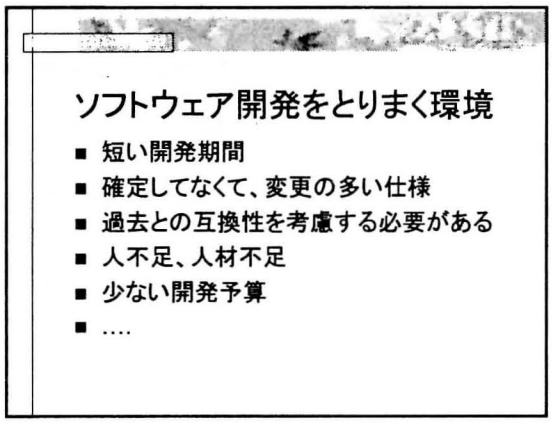
という話を聞くと寂しくなります。(しかし助けてもらっていますけど)

今後も、全ての方式は無理とは思うのですが、XPを実践してみたいと思います。

その中で、少しでも今の開発環境が改善できれば、いいかと思ってます。







XP eXtreme Programing

- 開発リスクを早期に軽減することを主眼におく
- 繰り返し型(iterative)の開発
- 開発中のコーディング、テスト工程に重点をおく
- 初期設計よりもリファクタリングによる再設計を重視
- すぐにでも始められるライトウェイトな方法論
- 「12のプラクティス」が示されている
- 「プログラマは人間である」の思想が流れている

リファクタリングすると

- 設計を見直すことができる
- プログラムを理解しやすくする
- バグを予防できる
- 変更に耐えるソースコードにする
- メンテナンスしやすくする
- プログラムを迅速に書けるようにする

XPの12のプラクティス

- 計画ゲーム
- 小規模リリース
- 比喩(メタファー)
- シンプルデザイン
- テスティング
- リファクタリング
- ペアプログラミング
- 共同所有権
- 継続的インテグレーション
- 週40時間
- オンサイト顧客
- コーディング標準

リファクタリングと設計

- 設計こそが重点ポイントで、コーディングは単なる組み立て作業
- 設計工程は順調←隙だらけの設計

- リファクタリングは事前設計の代わり
- 事前設計をある程度行い、リファクタリングしながら開発を進める

リファクタリング (Refactaring)

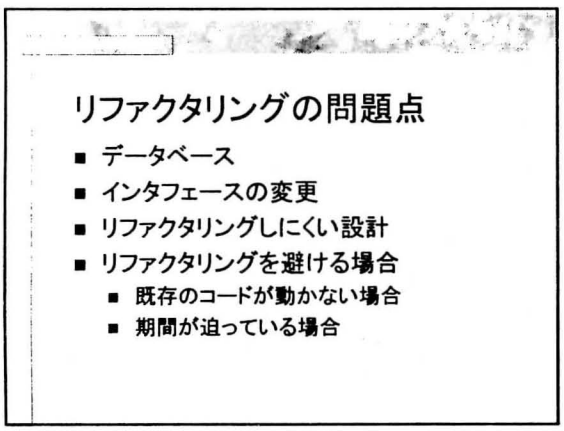
ソフトウェアの外部的振る舞いを保ったままで、内部の構造を改善する

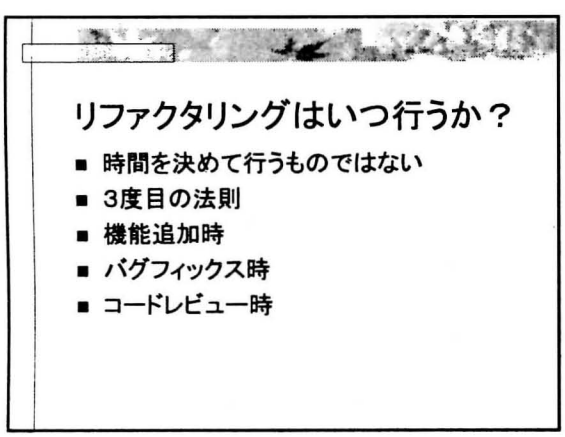
- 2重コードを取り去りる
- コミュニケーションを改善する
- 単純化する
- 柔軟性を加える

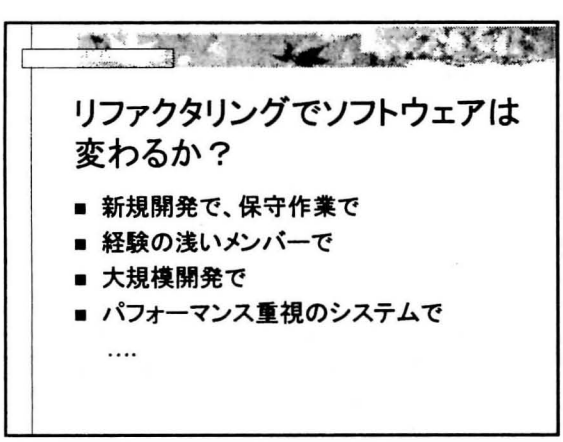
リファクタリングとパフォーマンス

- 時間分割して処理を行わせる
- パフォーマンスを意識してプログラミングを行う→×
- パフォーマンスのポイントを計測して見つけ、その部分をパフォーマンスを意識してプログラミングを行う

(最後にチューニングを残すのは不安である)


リファクタリングの問題点
■ データベース
■ インタフェースの変更
■ リファクタリングしにくい設計
■ リファクタリングを避ける場合
■ 既存のコードが動かない場合
■ 期間が迫っている場合


リファクタリングはいつ行うか？
■ 時間を決めて行うものではない
■ 3度目の法則
■ 機能追加時
■ バグフィックス時
■ コードレビュー時


リファクタリングでソフトウェアは変わるか？
■ 新規開発で、保守作業で
■ 経験の浅いメンバーで
■ 大規模開発で
■ パフォーマンス重視のシステムで
....

ソフトウェアの構造改革

－ 名古屋支部 ミニセミナーに参加して －

2001 年 12 月 26 日 熊谷章（PFU）

はじめに

SEA 名古屋支部の年忘れミニセミナーに参加した。話題が「ソフトウェアの構造改革」であった。司会の筏井さんが示したテーマは次の４つである。

1．コスト削減
2．期間短縮のための開発方法論
3．プロジェクトの管理の変更（プロセス）
4．仕様の変更にどう対処するか

昨年の記憶を頼りに書いているので、内容が多少違っているかもしれないが大意が合っていればよしとしたい。話題提供者が、与えられたテーマに対してそれぞれの見解を示した。主として、話題が「XP」に関与するものとなった。確かに、従来のソフトウェア開発のやり方と比較すれば、「XP」のそれは構造改革と呼ぶに相応しいかも知れない。リファクタリング、ペアープログラミング、繰り返し（Iterative）開発、プロトタイプ、小規模システムなどがキーワードとなり結構面白い議論が続いた。話題の核心を俺なりに理解すれば、従来の開発が属人性をなるだけ排除したものを目指したのに対して、「XP」は開発する人間中心のナイーブな仲間意識を共有したコミュニティを目指しているようにみえる。

XP とそれにまつわる話

これは、俺達が 15 年前から Smalltalk-80 を導入して以来ずっとやって来た開発方法でもある。「XP」では、主題がプログラミングであって、システムの分析や設計は作り上げたプログラムの実行を通して行われる。これは実に気分の良い方法である。１０年位前に、Smalltalk をシステム記述言語として、知的支援システムの開発を 30 名位のメンバーで 4 年間続けたことがある。外部機関のソフトウェア技術者が半分いた。このとき、最大の問題となったのが、外部機関の階層的な管理体系だった。たいていのソフトウェア会社では、課長が概念設計し、グループリーダが分析／設計し、プログラマがプログラミングをしていた。その工程は、ウオータフォール型だったので論理的に整合していた。しかし、「XP」のようなアプローチでは、プログラミングができない人はそのプロジェクトに参画できない。仕様確定、システムの分析／設計が実行されるプログラムを介して行われるので、従来の構造での課長やグループリーダは不要となった。われわれのプロジェクトの所為でソフトウェア会社にいろいろなトラブルが生じた。階層構造の組織なのに、それを考慮せずフラット構造を前提とした運営をわれわれが行ったからであった。いま、思い返せばおそれは正にソフトウェアの構造改革と言えただろう。

この時開発した「知的支援システム」は、すべて Smalltalk で記述され、仕様、機能、構造のすべてはプログラムとして記録された。目的としたシステムは、いまでいうマルチメディアの Hypermedia システムであったが、当時まだ HyperCard が製品化されていないときだった。従って、Hypermedia が正体不明であるため「知的支援システム」の評価はマチマチで面白かった。松下電器の専務はデモ開始後一分もしない内に言った。「よっしゃ、このシステムを丸ごと買え。技術、人を含めてなんぼするんや」。これには俺も魂消た。次に富士通の役員に見せた。「面白い。従来のものにないものがある。早速、関連部署の担当者を紹介するのでそこと今後どうすれば良いか検討して欲しい」。最後に PFU の社長らに説明した。「なにがいいかさっぱり分からない。今後どうする積もりなの」。この技術開発は、PFU の研究開発センターで 4 年間に渡り行われたもので、社内 60 名、外注 20 名の規模であった。この時の福次効果として、オブジェクト指向技術（Smalltalk-80）の導入、モトローラ 68000 を使用したワークステーション開発、Hypermedia 技術、Smalltalk-80 処理系の開発などがあった。これらも、ある意味ではソフトウェアの構造改革だったと思い返している。残念だったのは、最後のデシジョンメイキングが従来の方法でなされたために、「知的支援システム」が製品化されなかったことと俺達のプロジェクトが丸ごと松下電器に売却されなかったことである。この部分にも革新的なものが必要だったのであり、それを実行できなかった俺に責任があったと自責の念に苛まれている。しかし、このプロジェクトの所為で、日本の身近ににオブジェクト指向が定着し普及したとも言える。いま、Smalltalker として活躍している、青木淳や東田雅宏、加藤康人らはここから生まれた。

ソフトウェアの構造改革とは何をさすのか

話をソフトウェアの構造改革に戻す。テーマから俺が予想したものは、「XP」よりもっと社会的な性格を帯びたものだった。つまり、ソフトウェアの生産基盤や生産手段を変えるものを考えていた。具体的には、系列、ソフトウェア開発方法（How to）、プログラミングの重視、特化専用システムへの移行（ハードウェア、ソフトウェア、アプリケーション）、アプリケーションプログラミング言語開発などが念頭にあった。

産業構造としての系列

系列は、英語の単語として定義されるほどになった日本社会の仕事のやり方である。いまだに系列内の強固な連携が健在でこれが健全な自由経済活動を阻害している。コンピュータとソフトウェア技術は発展過程にあるので、技術競争こそがすべての活動の起動力になっているのに、仕事の需給が政治レベル、つまり系列関係で決まっていることが日本のソフトウェアの成長を妨げているようにみえる。この呪縛から抜け出てフェアーな自由競争が行われる社会の実現が構造改革に相応しいといつも思っている。では、どうすればよいか。市場を国内だけでなく、アジア全体や世界全体にすることが一つの解だと考えている。

ソフトウェア開発の How to

ソフトウェア開発の How to は、いつの時代でもソフトウェア機関の構造を変えてきたと思う。ソフトウェアの仕事は最初のうちは紙の上で、あたかも数学、物理、電気などのように論理式を書くことだった。カードパンチとコンピュータで実行させることは、コンピュータセンターの人に依頼して行われた。プリンタ用紙に印刷されたコンピュータの実行結果を見て、書いた論理式の誤りを見付け、目的とするプログラムを作成することが仕事だった。ソフトウェア技術者の仕事場には、机と椅子に紙と鉛筆と本があればそれでよかった。それが、いまでは机の上には数台のパソコンがあり、それらはしかもインターネットで世界中の情報源に繋がっており、24 時間コミュニケーションが可能になっている。ほとんどの場合、自分のパソコン上で目的とするソフトウェアの作成ができるようになった。しかも、多くのプログラムを利用できる環境が整備されている。これこそがソフトウェアの構造改革だと思う。これから先、この How to をどのように改革できるかが逆にわれわれソフトウェア技術者の最も重要な課題であろう。「あなたは自分の手で自分がやる仕事の道具とその実行環境を作ることができますか」という問いが一つの基準だと考えている。

プログラミングの重視と実際のプロジェクト

プログラミングの重視は、設計とプログラミングの関係を話題にしたいということだ。よく設計こそが重要でプログラミングはよい設計があればそれをプログラム言語にマッピングするだけの単純作業に過ぎないということを耳にする。しかし、ソフトウェアは最終表現形態であるプログラミング言語で表現され、その言語処理系で翻訳され、コンピュータ上で実行され、検証されない限り目的は達成されていない。モジュールの分割方法、データの定義、モジュール間のインターフェース、実行速度、システムの統合などは使用するプログラミング言語に大きく依存する。

現在のように、仕事のやり方が親、子、孫、ひ孫のように 4 階層におよぶ下請け方式が定着している中では、何が設計かが極めて難しいことになっている。従って、親レベルで要求される能力は、技術者ではなく、プロジェクトマネージメント能力である。原価率が重要なテーマであり、いわば政治や感情をも含めたコーディネーションが重視されている。従って、結果的にはツギハギだらけでよれよれのコンピュータシステムが社会で堂々と運用されていることになる。それらのシステムが事故を起したときだけ糾弾されるが、仔細にみれば事は深刻だということをソフトウェア技術者は知っているはずだ。ここに目をつぶってはならない。この現状を打破するには、プログラムの重視しかなく、設計との関係を深く考察し明解にすべきだと考えている。私見では、設計プロセスは、仕様、設計、プログラミングの三位一体の過程だと定義している。このアプローチの延長線上にアプリケーション開発言語がある。

専用システムへの移行

最近 WINTEL とは誰も言わなくなった。しかし、PDA 、パソコン、ノートパソコン、PC サーバのほとんどが WINTEL を採用している。最近は、組込みシステムでも同様だと聞いて驚いている。また、2001 年は、驚愕のニュースが続いた。日本国産コンピュータメーカが相次いで、スーパーコンピュータの製造中止、IBM コンパチブル汎用コンピュータの中止を発表した。儲からなくなったし、これからも儲からないから中止するという。後に、何が残るかいえば、WINTEL と Sun Microsysytems だけである。アメリカでは、IBM と HP が自前のコンピュータを持っているが、これらも今後は Intel のものを採用すると言明している。コンピュータがこのように変化すれば、当然ソフトウェアも必然的に構造的な変化を受ける。多くの人々が否応なく Windows で泣いているし、反 Windows として Linux などの Unix が使われているが、両者間のコミュニケーションが難儀なことで皆不自由を強要されているが、手も足もでない。これほどハードウェアの性能が上がり、価格が低下し、多くのフリーソフトウェアがあるならば、WINTEL だけにしがみつく必要がないのは自明だ。ある分野に特化した専用システムをソフトウェア技術者が作れる時代にもはやとっくに到達している。この実現もやはりソフトウェアの構造改革の一つだと思わずにおれない。

ソフトウェア構造改革が社会に課した問題はなにか

話の焦点がぼけてきたように見えるが、筆者はずっと「ソフトウェアの構造改革」について書いている積もりである。次はこのテーマを概念的なレベルで捕えてみよう。

21世紀の新しいビジネスモデル

社会にはその基盤として生産手段がある。それが、その上にある文化、つまり経済、政治、社会、法律を様々な形で規定していると考えられる。そして、文化のメタフィジカルな部分が理論とか考え方と呼ばれる。ソフトウェア産業もこの形態に当て嵌まる。基盤である生産手段がいつまでも人海戦術で属人性に頼っている限り、ソフトウェア産業の経済、政治、社会、法律などを含んだわれわれの日常生活は人海戦術の砂上に造られた楼閣に過ぎない。話題になるのが、経営資源である人、金、モノ、時間であるならば、そこでの定式化された価値基準は、リソースの投資効果としての利潤しかない。そして、ここでわれわれが目標としているものは、アメリカで実践され成功を収めたビジネスモデルでしかない。コンピュータにおける、そのビジネスモデルとは、ハードウェアとソフトウェアのモノ作りそのものではなく、夫々のアーキテクチャを独占することであった。現時点でそれは見事に成功し、全世界を制覇している。しかし、このビジネスモデルは２０世紀の寵児であったが、２１世紀では通用しないことが見え始めている。アメリカの極度の拝金主義に裏付けられた人、金、モノ、時間を要因とした企業経営のやり方だけでは先の展望がない。これに代わるものが必要で、それが広い意味での構造改革だろうと思う。

「情報」が生産、品質保全、再生の決め手

生産手段とメタフィジカルな考え方との間には双方向のコミュニケーションがあり、ある時には渾然一体となり新しいことを生み出すことがある。２０世紀に人類が発見した重要なものに、物理の量子論や相対論があり、これらは社会思想やいろいろな考え方に多大な影響を与えた。最近の例では「情報と構造の関係」の理論があげられる。ここでいう「情報」の概念とは、ものの構造を維持し、あるいは再生するものという意味である。ある構造が物理･化学的な因果論のみによって維持され再生されるものではないということが分かったのである。そのものの最初の発現は偶然であるが、いったん成立すればそれ自体として維持する性質を持ち自己を保全、再生、複製するものとして「情報」が使用されている。その具体例がDNA（情報システム）である。この発見は、革新的にみえる。ものを造るのに、「情報」が必要条件であり、物理･化学的なものは十分条件に過ぎない、という認識がわれわれの思考を刺激する。

ソフトウェア産業でモノ作りに必要なものは、人、金、物、時間であり、これは先に述べたものの物理･化学的なものに該当すると考えられる。では、ソフトウェア産業で、DNAに当たるものはなにか。それを見つけ出し、それを必要条件とした新しいビジネスモデルを展開することがソフトウェアの構造改革ではないかと考えた。コンピュータをめぐるほとんどあらゆるものが変貌を遂げつつある21世紀の最初の年にこのような事を考えるのもタイムリーで楽しかった。

おわりに

例によって深夜に「山本屋総本店」で、名古屋コーチンの味噌煮込みうどんと冷酒を堪能しながら、新しいソフトウェア時代の魁を感じていた。こんなことを考えさせてくれた名古屋支部の皆さんに心から感謝する次第である。

筏井　美枝子

ＸＰがどういった背景から生まれたのか、ＸＰとは何かを調べていたら、WEB 上の

Martin Fowler の“*The New Methodology*[*1]”

という記事にぶつかった。ＸＰを含む新しいソフトウェア開発方法論の流れが、わかりやすく説明されており、なかなか興味深い内容であった。

内容を少しだけかいつまんでご紹介しよう。

ソフトウェア開発の初期は方法論なしの時代（コードカーボーイ時代？）であり、物理的な効率だけが重視された。ダイクストラが構造化プログラミングを説いたころから、多くのソフトウェア工学的な方法論が生まれた。これらは、ウォータフォール型で上流工程を重視し、あらかじめ仕様や計画が明確である建築や機械工学をメタファとしたもの、つまり Monumental（または heaviweight）な方法論であった。

これに対して、新しい方法論の流れは、仕様や環境というのは常に変化するものであるとし、変化を前提に機敏に適応しようというもので、Agile（または lightweight）な方法論と呼ばれる。この Agile な方法論のもう一つの特色は、管理的な色彩の強い従来のやりかたから、関係者の協調や想像性を開発の主体にすえた点である。こういった考えは、オープンソースの流れにもみえ、開発者を単なる人月や部品のようにみなす従来の管理的な概念に対して、高らかに人間宣言をしたような感があり、これだけでも技術者が大いに共感するだろうという気もする。

この文章の中で彼は、Lightweight Methodology が正しいとか、それを使いなさいといっている訳ではなく、自分の開発しようとしているシステムがどんなタイプの開発なのかによってふさわしい手法を選択し、それを自分の開発に合わせて改善していくことを提唱している。

そして、Lightweight Methodology として、ＸＰ、Crystal Family (by Alistair Cookburn)、オープンソース、SCRUM、Feature Driven Development (by Peter Code)[*2] などをあげ簡単な説明を加えている。それぞれにリンクもはってあるので、興味のある人は読んでみると面白いと思う。彼は、現在ＸＰが人気を博しすぎて、他の手法が影に入ってしまっているのはむしろ残念なことだともいっている。

Fowler のおかげで、ＸＰという方法論の背景や、位置づけがだいたい見えたところで（実は、この時点で、ＸＰ以外の方法論、Crystal Family や Feature Driven Development にもかなりの興味がそそられてきていたが、この辺はあっさりと概要にとどめ）、ＸＰを詳しくみてみることにする。

ＸＰについては、かなり本がでているし、WEB 上にも多くのサイトがあるので、説明するまでもないのだが、具体的なプロセスの全貌をつかむ意味で、当日は、http://www.extremeprograming.org から、XP のプロセス図を借り、それを白板に投影させ、上から具体的なポイントを書き込みながら、プロセスをたどっていった。しかし、今思えば、ごちゃごちゃとした図になって、かえってわかりにくくなったかもしれない。

かいつまんで特色だけをいうと、ＸＰは２～12 人の小規模の開発で、１～３週間ほどの開発を Iteration（繰り返し）することにより、仕様の変更、コストや工程の見積もり調整に柔軟に対応している。

従来のような設計やドキュメント作成は行わない。設計は、CRC カード、ペアプログラミング、リファクタリングなどにより、適時、仲間とレビューしながら行う。従来のドキュメントは、テストプログラム（単体とアクセプタンス）やソースコードがこれに対応する。また、テストを重視している点が大きな特色といえる。プログラム作成の前に必ず単体テストプログラムを作成することで、そのプログラムの仕様が明確になる。さらに、プログラムソースと必ず対でリリースすることで、次の開発の基盤（品質保持）をになうしくみになっている。

さらに、プロジェクトに顧客を加え、常に協調しながら開発を進めていく点も特色である。

以上のおおまかな特色からいっても、また lightweight という名称からいっても、大規模システム、safety-critical なシステム開発、そして工学的な品質を追求する場合には、こういった開発方法論が向かないことは明らかである。

ゆえに、lightweight な方法論とは、ある条件での開発方法論であり、条件を満たせば効果があるという位置づけをして終わっても良いのかもしれない。

一方で、こういった方法論の新しい流れはいったい何を意味しているのかを考えてみると、また面白いことに気がつく。

元来、世界（システム）は複雑であり、常に変化するものである。その一部を切り取り、シンプルに多少でも安定した形で捕らえようとして、これまで様々なモデリング技法や開発方法論が提唱されてきた。

それらは、けっして森羅万象に対応するものではないはずなのに、いつのまにか、世界や思考のほうを既存のやりかたにあわせてしかるべきといった場面が増えてしまう。

捕らえようとする世界の性質、複雑さ、変化の速度は、時代とともに当然変わってくる。となれば、それに合わせた方法論が求められてくるのは自然の流れなのではないだろうか。そう思うと、lightweight な方法論とは、情報化時代に求められた方法論であるといっても良いのかもしれない。

また、ドメインによっても方法論は違うはず。とすれば、これからは、多様な方法論の時代に入っていくのかもしれない。

(*1)：http://www.martinfowler.com/articles/newMethodology.html

(*2)：http://www.togethercommunity.com/coad-letter/Coad-Letter-0070.htm

XPはHYPEか？
― これからのソフトウェア開発環境・プロセスのありかたをさぐる

SEA 名古屋支部
2002年 新春討論
2002.1.22

筏井 美枝子

22 January 2002 1

新しいソフトウェア技法（その1）

- ***Martin Fowler***

"The New Methodology"

http://www.martinfowler.com/articles/newMethodology.html

22 January 2002 2

新しいソフトウェア技法（その2）

- **From Nothing,**
  to Monumental(heavyweight)
  to Agile(lightweight)

The new methods attempt a useful compromise between no process and too much process.

22 January 2002 3

新しいソフトウェア技法（その3）

- agile methods :
  - *adaptive rather than predictive*
  - *people-oriented*
    rather than process-oriented

22 January 2002 4

新しいソフトウェア技法（その4）

- ***adaptive rather than predictive***
  - predictive（予見的）
    the traditional engineering metaphor（建築など）
    Construction
  - adaptive（適応的）
    unpredictable Design（ソフトウェア開発のほとんどがデザインと言える）
    The Adaptive Customer（顧客との密接な関係）

22 January 2002 5

新しいソフトウェア技法（その5）

- ***people-oriented***
  rather than process-oriented
  - *process-oriented*（手続き中心）
    The notion of people as resources（開発者＝人月）
  - *people-oriented*（人間中心）
    the developers must be able to make all technical decisions
    （現役の開発者を信頼して任せる）

22 January 2002 6

新しいソフトウェア技法（その6）

- ***adaptive***
  - adaptive processes （適応型プロセス）

    Iterative development（繰り返し型開発）
  - The Self-Adaptive Process（自己適応型プロセス）
    - *What did we do well?*
    - *What have we learned?*
    - *What can we do better?*
    - *What puzzles us?*

22 January 2002 7

新しいソフトウェア技法（その7）

- ***The lightweight Methodologies***
  - XP *(Extreme Programming)*

    Kent Beck 1996年 クライスラーC3プロジェクト
  - Crystal Family

    Alistair Cockburn the least disciplined methodology
  - Feature Driven Development *Peter Code*
  - SCRUM
  - オープンソース

22 January 2002 8

新しいソフトウェア技法（その8）

- ***What should we use ?***
  - predictive process（予見的プロセス）
    - *50人以上のチーム*
    - *開発者の質が低くやる気がない*
    - *固定スコープ、固定契約*
  - adaptive process（適応的プロセス）
    - *不確実で流動的な仕様*
    - *責任感があり、やる気のある開発者*
    - *話しがわかり、一緒にやってくれる顧客*

22 January 2002 9

新しいソフトウェア技法（その9）

- ***If we choose the adaptive process which should we use ?***
  - XP
    - *2～12人の開発者がみんな乗り気だったら*
  - Crystal Family
    - *規律というものがこれまでなく、人数も多かったら*

You have to take change of your process, monitor it , adapt it to your circumstances

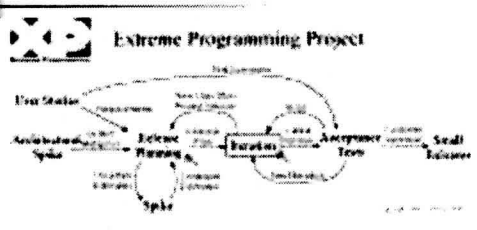
22 January 2002 10

XPとは

- ***XP is a lightweight methodology***
  - *2-12人の小,中規模チーム*
  - *仕様が決まっていないまたは仕様変更の多いシステム向け 短期間の繰り返しにより、just in time の開発*
  - *革の視的な方法論。開発者自らがが導入しようとするのが普通。*

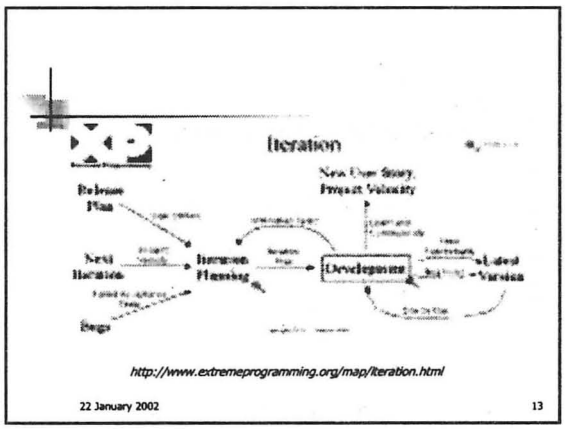
http://www.extremeprogramming.org
http://c2.com/cgi/wiki?ExtremeProgramming

22 January 2002 11

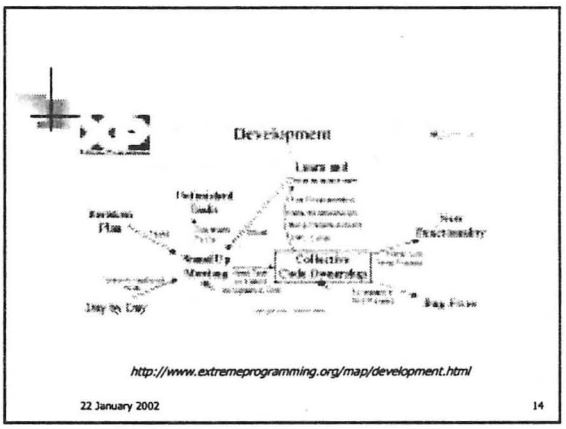


http://www.extremeprogramming.org/map/project.html

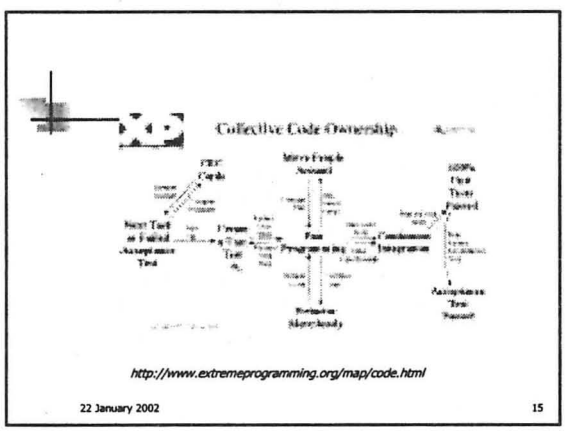
22 January 2002 12



http://www.extremeprogramming.org/map/iteration.html



http://www.extremeprogramming.org/map/development.html



http://www.extremeprogramming.org/map/code.html

XP事例 1 (その1)

- ***Launching Extreme Programming at a Process-Intensive Company***

  James Grenning, Object Mentor

  (November/December, 2001, IEEE SOFTWARE)

- **System** :*an embedded-systems application , safety-critical systems running on Windows NT part of a network of machines that had to collaborate to provide services.*
  大規模システムの中の1サブシステム、
  他のグループは従来のプロセス(標準的なウォーターフォール型)
- **programming language**:*C++*
- **team size** :*顧客1人に開発者2人*

XP事例 1 (その2)

- **既存のプロセス**:
  オーバーヘッド/デッドラインがきつい。
  技術者は逃げ出してしまい、
  要求仕様は部分的にしか定義されていない。
- **カルチャ**:
  ROSEでの要求仕様/設計、レビュー、承認。
- **XP導入への反対意見**:
  実現は危険である
  うちのカルチャには馴染まない
  ドキュメントがないのは困る。 -> ユーザストーリはユースケースに。
  safety criticalなプロジェクトなので、レビューは欠かせない。
  -> Iterationの中でレビュー
  その月の設計上の決定事項をドキュメントにする

XP事例 1 (その3)

- 実践
  - **設計**
    顧客からの要求仕様書をカードに書いて、ユースケースと名づける(数日で、125枚)。
    白板に設計図をスケッチ(CASEツールは時間の無駄)。
    自動テストとリファクタリングで設計を進化
    コーディング規約は他チームとのインタフェースで部の規約を使う(コメント)
  - **開発**:
    ペアプログラミング。
    6個のクラスから始まって、約100個のクラスを実現
    ユニットテストによりさまざまな副作用を防ぐことができた
    エラーはすぐにFIX
  - **結果と評価**:
    生産性と品質結果をマネージャとメンバが評価 -> 新たなパイロットプロジェクト2つ

XP事例２（その１）

- ***Using Extreme Programming In a Maintenance Environment***

  Charles Poole and Jan Willem Huisman, Iona Technologies

  - 背景 :*Corbabased middleware product called Orbix,*
    コードは複雑で膨れ上がる
    仕様は変化しやすい
    時間要求は厳しい
    エンジニアは修正に継ぐ修正で常にプレッシャー
    テストの効率が悪い

22 January 2002 19

XP事例２（その２）

- **Extreme maintenance**

  プロダクトがライフサイクルの主流にあるうちは、XPモデルに従う。

  - XP導入:
    インデックスカードに作業を記録し、ストーリーボード（バグレポート・バグFixストーリ）
    2週間単位の作業にする。多ければストーリを分割
    毎日15-30分のスタンダップミーティング
    自動テスティング（夜）
    ペアプログラミング
    リファクタリング
  - 効果:
    ビジビリティが良くなった。
    ストーリーボードで他の人の進捗もわかる。
    モチベーションが非常に高くなる。
    ペアプログラミング -> コードベースのクロストレーニング
    見積もり: 先週やったと同じくらい今週もやると仮定（Yesterday's Weather）

22 January 2002 20

まとめ

- システムのタイプによって開発方法論は変わる
  - 人数、要求仕様の変更頻度、ドメイン(safety critical)など
- 流れはagileな開発方法論に
  - 予見的/官僚的なプロセスから、適合的/人間中心のプロセスへ
- XPはある条件を満たす開発にはきわめて有効
  - 少人数、要求仕様の変更が多い、顧客が協力的、
    そして、技術者が前向きである場合
- XPとCMMは対極(weightと)のものではなく補完しあうもの？
  - どうやって補完するかは今後の課題

22 January 2002 21

さいごに（その１）

- 開発方法論の流れに思う
  - 世界（すべてのシステムやプロセス）を記述できる方法論や、
    モデル言語があるはずだと思うのは間違い。
  - すべての方法論やモデルは世界を理解するために恣意的に考えられたもの。
  - 元来、世界は複雑で常に変化するもの。
    それを少しでもシンプルに、安定した形で捕らえる技術が要求される。
  - 捕らえようとする世界の複雑さや変化の速さ、そしてその世界に住む人間の価値観に合わせて技術も変化する。
    agileな方法論は、今の情報化社会が必要としているもの？

22 January 2002 22

さいごに（その２）

開発方法論の流れに思う

- 一方で、その方法論によって作り出されたプロセスやシステムもまた
  世界の一部である限り、世界もそれによって何らかの影響を受ける。
- monthly の変化に対応したプロセスは、weeklyの変化を生み、
  weekly の変化に対応したプロセスは、dailyの変化を生むとしたら....
  低価格、ラピッド開発を求めるスパイラル（？）は、どこに行き着くのか？

22 January 2002 23

遠隔学習の概況

― アメリカの大学を中心に ―

黒澤 公雅
(宇都宮ビジネス電子専門学校)

本稿は，SIGEDU (教育分科会) の２月月例会で話題として取り上げた話をまとめたものである．

まず，SIGEDUのMLで流した月例会の案内文をそのまま引用すると，以下の通りである．

インターネットの普及とその後のブロードバンド化は，ネット上の教育を現実的なものに変えつつある．日本においても，文部科学省の省令改正を受けて，遠隔学習での学士や修士の取得が可能になっている．実際に専門学校で，通信教育による学士取得 (いわゆるダブルスクール) を指導している立場から，従来の通信教育の問題点を整理するとともに，遠隔学習の現状について調べてみた．

国内外，特にアメリカの大学・大学院について，Distance Learning Program (Peterson's) のガイド等から作成したデータ(287校，付表リスト参照) をもとに，まとめを試みた．日本の通信制大学院や，ネット授業のホームページについても触れたい．日本で自宅にいながらにして，本当に海外の大学の学士・修士が取得可能か，考えてみるのも面白いと思う．

この分野で，より多くの議論や報告書提出が，日本全体の嵩上げにも役立つだろうと考えて書いている．目的は，国内外の遠隔学習について，知見を深めることである．具体的には個々人により，

ビジネスの方向性を考える (遠隔学習でビジネスを行う)
自分の仕事にする (遠隔学習で教えてみる)
自分の学習に役立てる (遠隔学習で学位を取得する)

等である．

１．定義

Guide to Distance Learning Programs によれば，遠隔学習とは，「ケーブルテレビ，衛生放送，オーディオテープ，ファクシミリ，パソコン通信やインターネット，テレビ会議システムなどの通信技術を利用して，キャンパス外の学生に教育プログラムを提供すること」，または単に「オフサイトにいる学生にも教育プログラムを提供すること」とある[1][2]．ここでは，従来の通信教育を「狭義の意味での教科書ベースの通信による教育」と捉え，これに対する遠隔学習を「最新の通信技術を利用して行われるオープンキャンパス学習」または「単に教科書を郵送するだけではない通信教育のこと」と定義し，明確に区別して考えることにする．

通信教育は，米国では Independent Studies や Indivisual Studies といい，その他の英語圏では Continuing Studies や Off-Campus Studies という傾向がある．遠隔学習のほうは， Distance Learning 以外に Correspondence Cource や External Program といった表現が使われるときもある．

一般には，通信教育が発展した形が遠隔学習であり，両者は必ずしも明確に区別されているわけではない．どちらも「大学に通うことなく，単位や学位を取得できる
(広義の) 通信教育の一種」と考えることはできる．その他，e-learning や WBT(Web Based Training), CBT(Computer ....), TBT(Technology ...) も，遠隔学習とほとんど同じ意味で使われていると考えてよいであろう．

２．背景

では，遠隔学習またはeラーニング(e-learning) に対する需要はどのくらいあるのだろうか．

図１は，日本に限った話だが，eラーニングの市場予測の一例である[3]．２年で２～３倍は，この低成長下に幾何級数的成長が見込める数少ない分野の一つであることを示している．米国など海外市場でも，２年で２～３倍という予測がある (月例会では現状把握のため，米英仏と日本の教育改革の

違いについても資料を見ながら考えたが，ここでは立ち入らないことにする).

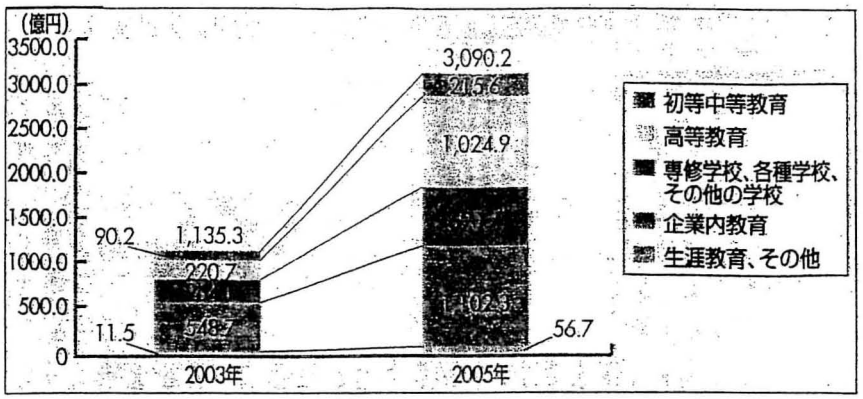


図１ e ラーニング市場の推移

図２の「日本の人口ピラミッド」は，最近よく引き合いに出される図である[4]. 4割強が大学に進学し(まだ6割もマーケットがあると見る人は少ない)，少子化による教育機関の危機が叫ばれて久しい.

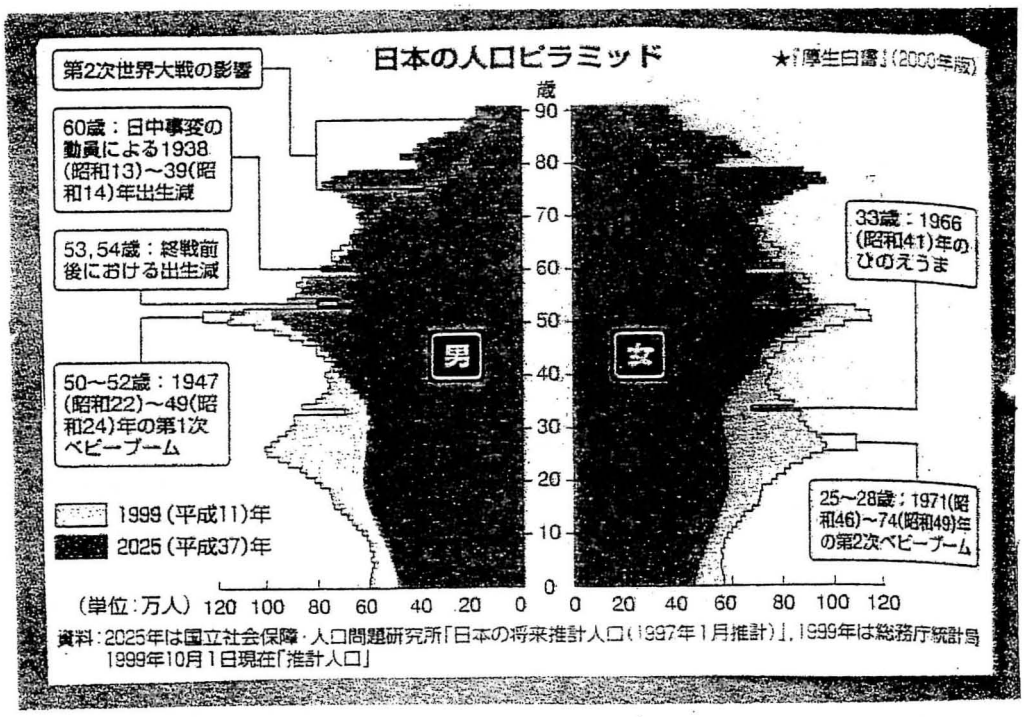


図２ 日本の人口ピラミッド

その内訳を図式的に示したものが図３であるが，少し視点を変えて見てみよう[5]. 1999年の調査によると，通信教育の大学生が14万人，放送大学の学生 7万人をこれに加えると，通信教育または遠隔学習を受けている大学生は計２１万人にものぼる点に注目する．大学生 259万人の実に一割弱である．つまり，通信教育や遠隔学習は，いまや特殊なスタイルではないことがわかる．さらに，社会一般に提供されている各種の学級・講座の受講者が，1,300万人・1,500万人といった数にのぼること，図２で，40歳代 50歳代の人口が 30歳以下の全人口に匹敵することを考えあわせれば，社会人を中心とした教育の潜在需要が見えてくる．この需要に答える方法の一つが遠隔学習である．

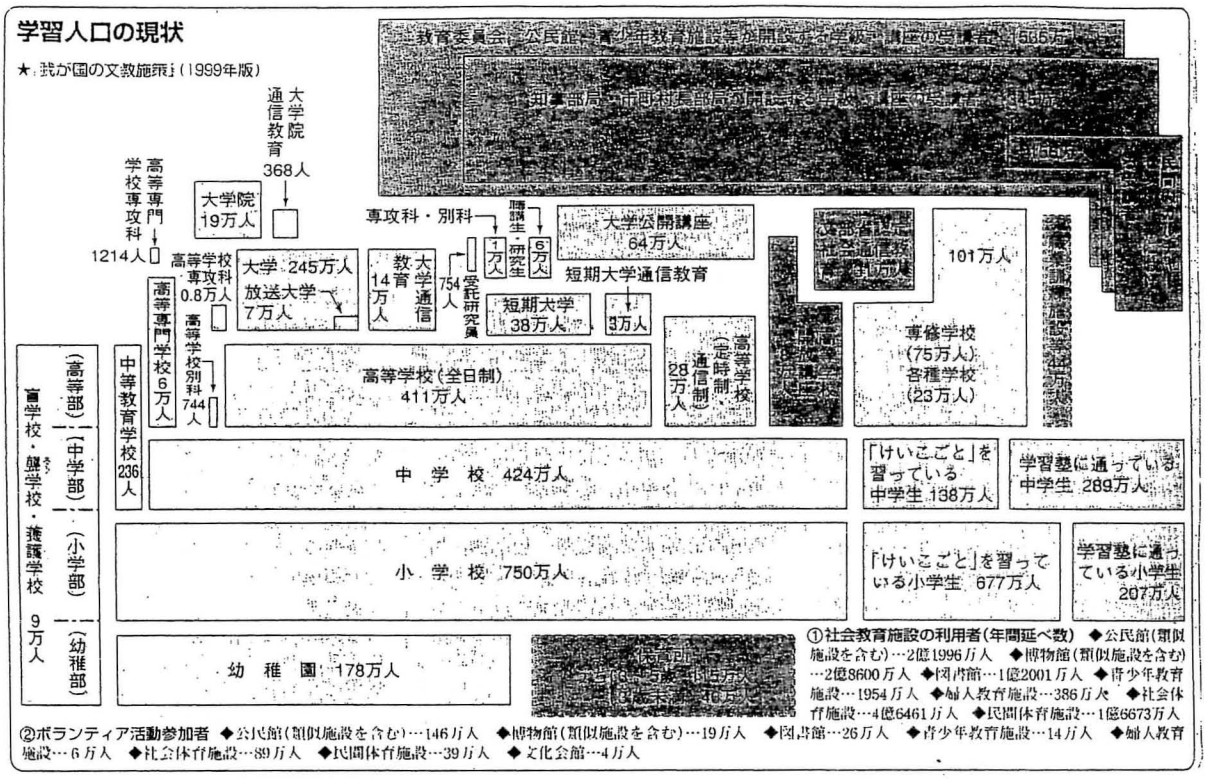


図3 学習人口の現状

3．通信教育の歴史

ここで簡単に歴史を振り返っておこう．遠隔学習の原型は通信教育である．

歴史をひも解けば，日本では江戸中期，本居宣長が旅行中の賀茂真淵と出会い（「松坂の一夜」として有名），「古事記」研究を志し，1763年から6年間書簡による教えを受けたという記録がある．これが，日本の通信教育第一号ではないかと考えられている[6][7].

一般的解釈としては，1840年英国で近代郵便制度の確立を受けて，Issac Pitman という人が速記の通信教育を始めている．これが，実質的な世界第一号であろう．日本でも，1871年 (明治 4年)，官営の近代郵便制度が発足し，さらに 14年後の1885年 (明治16年) 5月，鳳文館が経史詩文の通信教育を始めている．(日本の実質第一号) 同年 9月には東京学館が校外部を設置し，簿記と速記の通信教育を始めている[7].

しかし，驚くべきことに，通信教育が大学教育にはじめて導入されたのは，日本の近代郵便制度発足よりはるか 13年も前の 1858年のことであり，場所はロンドン大学である．日本は江戸末期，まだ飛脚が走っていた頃のことである．この有名な英国の国立大学は，通信教育の草分け的存在であり，その後，一世紀半に渡り，キャンパスに通えない学生に対し，全世界で通信教育 (いまでは遠隔教育) により卒業生を輩出し続けてきた[8]．余談だが，ロンドン大学が通信教育を始めた本当の理由は，植民地政策のためと推測される．充分な調査はしていないが，海外在住の英国人やその子息の高等教育の需要に答えるためだったと考えるのが自然であろう．米国以前に，英国が教育の面ではグローバル化の道を歩んでいたといえよう．

日本に話を戻すと，戦後1947年(昭和22年) 学校教育法で通信教育が初めて正規に公認される．その後，実に半世紀をへて，1998年3月 大学院設置基準の一部を改正する省令をもって，遅ればせながら修士課程でも通信教育を公認する (つまり，それまでは日本の大学院を通信教育で出ることはできなかった)．さらに，2001年3月の改正省令で，通信制大学の全単位を放送やネットで取得しても卒業単位として公認されることになった．これによって，124単位中 30単位以上はスクーリング (面接授業：つまり，教室で行う普通の授業) で取得せねばならぬとしていた従来の制限が撤廃された．簡

単にいうと，完全バーチャル大学でも正規の大学として成り立つことになった (バーチャル大学がバーチャルな存在ではなくなったわけである)．日本でも，遠隔学習を最高学府で実施するための制度がようやく整いつつあるといえる．

4．通信教育の問題点

(1) 孤独感．
(2) 進捗管理がむずかしい(自己管理)．
(3) テスト受験に指定された場所に行かなければならない．
(4) スクーリング(面接授業)に参加しなければならない．
(5) 結果的に脱落者が多くなる．

現実の細かい話をすれば，特に(4)は，金曜 (または月曜)を含む週末の３日間に実施されることが多く，社会人の場合，授業スケジュール通りに休みをとることが難しく，大きな障害の一つになっている．これに対し，大学側もたとえば産能大学の場合，iNetCampus という遠隔学習 (インターネットによるゼミかビデオオンデマンド) を取り入れ，遠隔学習かビデオ学習１日とスクーリング２日間で３日分のスクーリングとみなしたり，平日の夜間に１０週連続で短時間単位でスクーリングを実施する等，さまざまな工夫を行っているが，抜本的な解決にはなっていない[9][10]．

また，実際にダブルスクールで大学の単位取得を指導している立場でいうと，恐らく教師や学校事務局のサポートがなければ，大半の学生は脱落していると推測される．併修校 (専門学校で大学と提携し，大学の単位を同時履修させる制度)ができる前の純粋な通信教育による学位取得率は約 4% というデータもある．現在でも，日本の通信教育の場合，卒業率は 10% 以下の大学も多い．サポート体制の充実が，如何に大切な要件かということがわかる．遠隔学習になっても，(1)(2)(3)の問題点は残る．

5．米国のガイド分析 [1][2]

さて，e-ラーニング市場が伸びており，また，規制緩和で日本の大学も遠隔学習に参入できる条件が整った (実際，参入している)ことはわかったが，では，遠隔学習のメリットを本当に我々は享受できる状況にあるのだろうか．

月例会では，日本の通信制大学院，次いでカナダ，英国，豪州など世界の代表的な通信教育または遠隔学習の大学にふれた後，米国の遠隔学習大学の話に移ったが，ここでは，全体的な印象を述べるにとどめ，米国のデータの紹介に進むことにする．その印象とは，英国を中心とした欧州の大学では，従来の伝統的な通信教育の手法を守る傾向が強く，一方北米，特に米国では，インターネットに限らず新しい技術を積極的に取り入れる傾向があるように感じられたということである．豪州はその中間である．大学の歴史の長さに逆比例するというか，文化の違いのようなものが大きく影響していると考えられ，私には興味深く感じられた．

さて，何といっても数が多いのは米国である．米国では，かなり多くの大学が古くから通信教育を実施しているが，1990年代に入ってさらに多くの大学が通信および遠隔学習市場に参入した．学士だけでなく，修士・博士も取れるところが多い．遠隔学習だけでは完全に卒業できないところも多いが，それでも常時通う必要はない．その傾向は，ますます進む方向にある．このあたりを少し数量的に確かめようと思った．

今回まとめたデータは，1998年版の Guide to Distance Learning Programs をベースにしており，米国の大学が中心である．データは97年頃のものである．少し古いデータであるからその点は承知して見てほしい．実は，調べ始めたとき

(1) 98年版しか詳しい資料が手許になかった．
(2) その時点で同等に詳しいデータが得られそうな最新の資料が見つからなかった．

ことによる(後に詳しい02年版も入手している)．取りあえず，4年程度前の入学者数に対する学位取得者数の比較の意味でも98年版は意味があると考え，データ作成を試みた．学位を取得 (卒業)でき

る可能性がどのくらいあるかは，入学を検討する者にとって最大の関心事に違いない．残念ながら，卒業者数のほうのデータ入手が難しく，まとまっていない．少なくとも，4年以上前から遠隔学習を実施している点で，データのもとになった大学は実績面で検討しやすい大学ということはいえる．

98年版の上記ガイド約 800校から，以下の条件のもと，287校を抽出した
(英国 2校，カナダ 15校を含む．ロンドン大学は原書からして当時遠隔学習大学とはみなされておらず，英国 2校には含まれていない)．

＜抽出条件＞
(1) 2年制の大学を除く(ただしB(学士)を取れるものは例外)．
(2) 学位はB(学士), M(修士)とはっきり記載されているものを選択．
(3) 明らかに地域限定の記載のあるものも除く．
(4) 明らかに遠隔で準学士しか取れない 4年制大学を除く．
(5) 特別な大学院修了証書プログラムは修士 (M)から除外．
(6) B,Mは，一コースでも該当すれば入れる．

抽出した 287校に対して，定量化できるパラメータを考え，主にガイドを拾い読みしながら，種々の条件や数字の有無を調べていった (全データ数は約15,000件弱)．判断が難しく，データに含めてあるが，宗教系の母体を持つ大学が 56校含まれている．中には，特定の宗派に属していることを入学要件とする大学もある．しかし，宗教系の大学を全部除いても，米国だけで 200校以上候補が残る．同じ基準で考えると，02年版のGuide to Distance Learning Programs に掲載されている大学約1100 校中，比例換算で米国だけで 300校近くが候補として残ることになる．

図 4 は，どの辺りの大学が多くアクレディット (認可) を受けているか，地域的なバラツキがあるのではないかという発想から描いてみたものである．

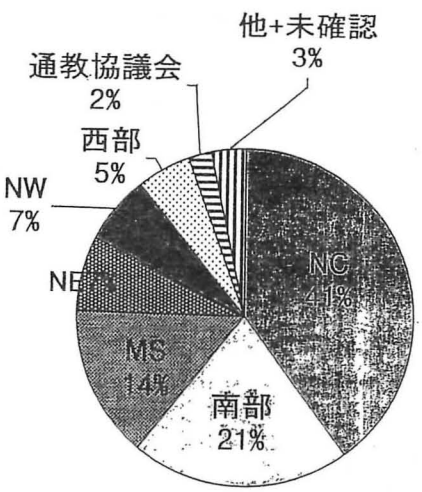


図 4 米国 (抽出 270校) のアクレディット

米国にはアクレディットの問題がある．たとえば，カリフォルニア州などでは，いとも簡単に大学が作れてしまう．要するに，事務所に机1個しかない大学もある．そんなところの出す学士，修士，博士に本質的な価値はない．よって，きちんとした認可団体が必要で，米国では米国教育省 (U.S. Department of Education) または高等教育許可委員会 (Council for Higher Education Accreditation) が認めた，以下 6つの地域別の組織 (認可団体) が存在する (カナダにも同様の問題があるが，こちらは各州政府が州内の大学をきちんと評価している)．

NE=The New England Association of Schools and Colleges
MS=Middle States Association of Colleges and Schools
NC=North Central Association of Colleges and Schools
NW=Northwest Association of Schools and Colleges

南部=Southern Association of Colleges and Schools
西部=Western Association of Schools and Colleges

この他，通教(=通信教育.・研修協議会) というのもある．地域内の各大学は，認可団体に対し，コース単位または大学全体として所定の要件を満たすか自己評価 (self-study) して認可申請し，厳しい審査をへて認可 (アクレディテーション) を得る．これが，対外的に大学の信用につながっている．認可は，通常 5年から 10年の周期で定期的に行われている．無論，日本では文部科学省が大学の設置時に認可を行っているので，こんな問題はないわけだが (実はそんな日本でも，1998年10月の大学審議会答申をへて，大学にたいする第三者評価機関が設立されている)．

さて，図４であるが，このグラフからだけでは何ともいえないし，よそ様の国のどの辺りにはいかがわしい大学が多いとか調べることには抵抗がなくもないが，上記のように，日本とは実状が異なるので注意は必要である． NC は，米国北部から中部にかけてのかなり広い範囲の州をカバーする団体で，月例会ではこの辺りは人口希薄で，家と家とが非常に離れている地域が多いといった興味深い話もあった．遠隔学習に対するニーズが多い地域ではある．

図５～８は，受講場所に関するグラフである．図５より，自宅のみで受講できるところが22% (1/5強) もある．図６，図７は，図５の補足である．場所毎の比率を表した図８を見ると，まず職場が多い (62.0%もある)． これは，米国では会社単位の教育ビジネスが成り立っていることを示す．スタンフォード大学のように，会社単位の受講を条件とするところもある．基地 (35.9%) も思ったより多い．徴兵下でも基地の敷地内で勉強できる体制が整っていることが推測できる．オフキャンパスセンター (23.0%：off campus center) とは，キャンパス外での授業実施場所を指す．米国では，専用線などを利用して，主に大学が立地する州内に複数のサイトを設置し，そこに学生を集めて遠隔学習を実施するというパターンがある．オフキャンパスセンターを持つ66校の教育場所 (キャンパス外) の数をグラフにしたものが図７である． １～４ヶ所くらいが多い．20ヶ所とか，多いところでは100ヶ所も用意しているところもある．一般にカナダの大学は多く，たとえば Acadia Univ. ：20ヶ所， Univ. of Guelph ：56ヶ所， Univ. of Calgary ：95ヶ所 といった具合である．米国でも，たとえば Southwest State Univ.(ミネソタ州) ：43ヶ所， Univ. of Southern Indiana ：200ヶ所以上 と多いところもあるが，例外である．

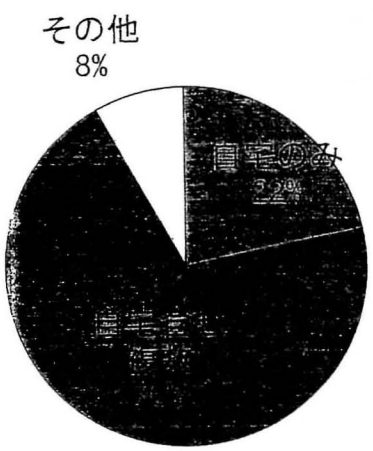


図５ 受講場所 (遠隔学習 287校)

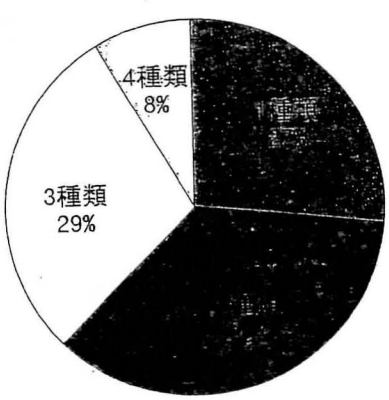


図６ 自宅外の受講場所 (自宅を含む複数の場合，202校)

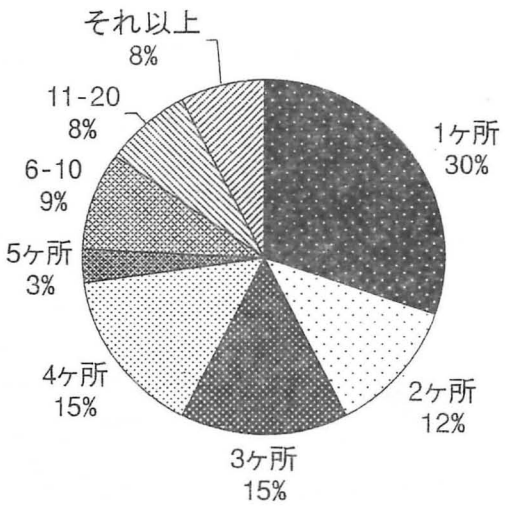


図7 オフキャンパスセンター (66校)

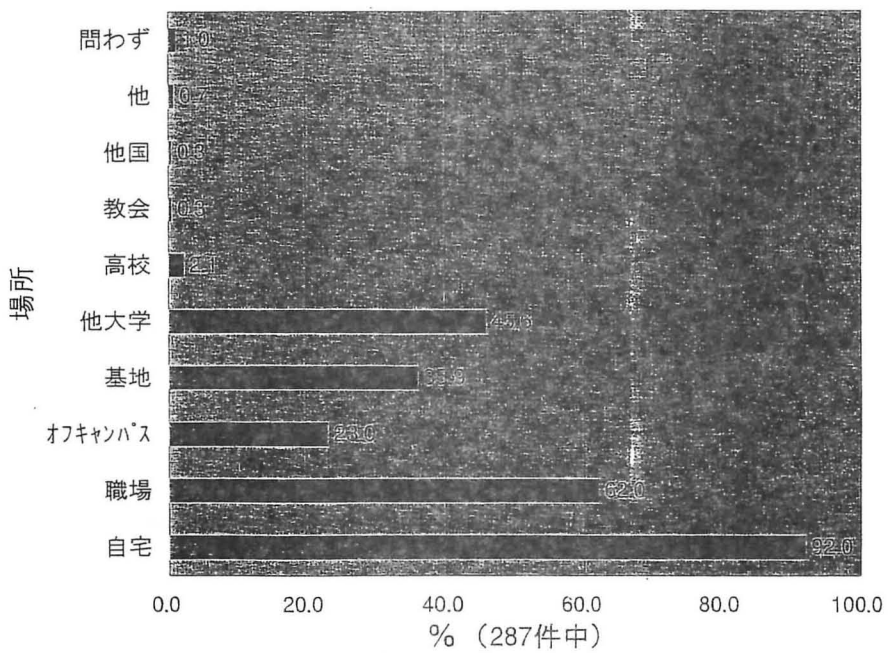


図8 受講場所 (287校)

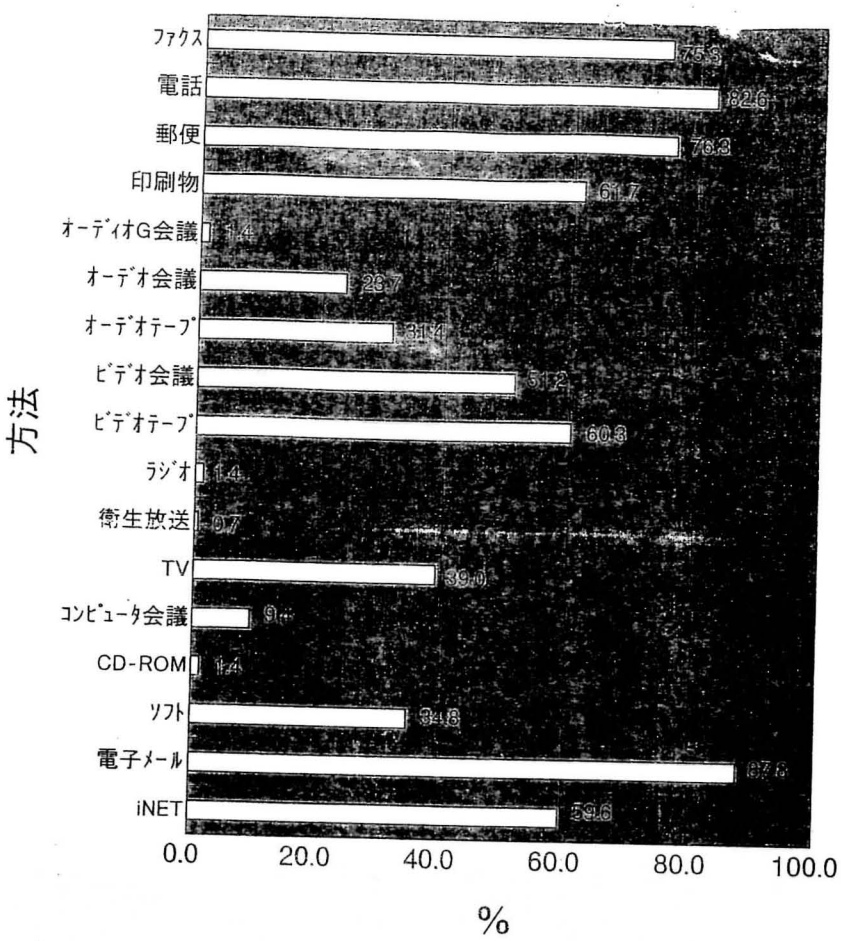
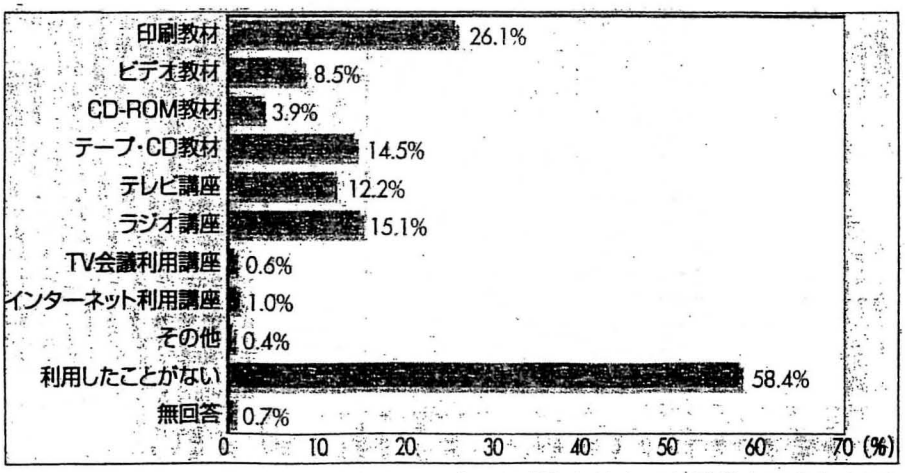


図９ 受講方法 (287校)



（ALIC「遠隔教育の利用動向調査」全国）

図１０ 利用したことのがある遠隔教育と教材 (日本)

受講方法は図９のとおりである．図１０の日本のデータ (2001年版)と比較すると， 97年の米国のほうが断然先を走っていることが一目瞭然である．[11] 日本は，日本語という独特のエリアであるため，席巻されずに済んでいるともいえる．図９の各項目の意味は，以下のとおり．

ビデオ会議は，衛星放送，光ファイバ，CATV等による双方向TVをすべて含む．オーディオ会議は，複数の人が同時に会話できる電話回線等を利用した音声会議を指す．日本でも3人同時に話すことは，いまのシステムでも可能である．オーディオG会議とは，オーディオグラフィック会議のことで，(私自身見たことはないが) 電話回線を使用しても毎分 1枚くらいの画像情報なら (要は，リアルタイムでなければ) 送れるので，専用の送受信機があれば議論や説明には使えるはずである．そのようなものではないかと考えているが，知っている人がいたら教えてほしい．ソフトとは，ここでは自宅のパソコンにインストールして使う学習教材 (ソフトウェア) をいう．遠隔学習を標榜する大学ではWebの利用はいまではほぼ100%であろうが，いずれにしても 97年の時点で，ビデオ会議 (51.2%)，オーディオ会議 (23.7%)，TV (39.0%)など，様々なメディアが非常に高い割合で使われていることがわかる．

次に，米国の遠隔学習で，どのレベルの学位がどのくらい提供されているのだろうか (どのくらい取得可能かではない)． 図１１は，以下の基準で作成した種別の学位提供率である (現在の日本では，修士・博士だけでなく，学士も学位の一つとみなされている)．

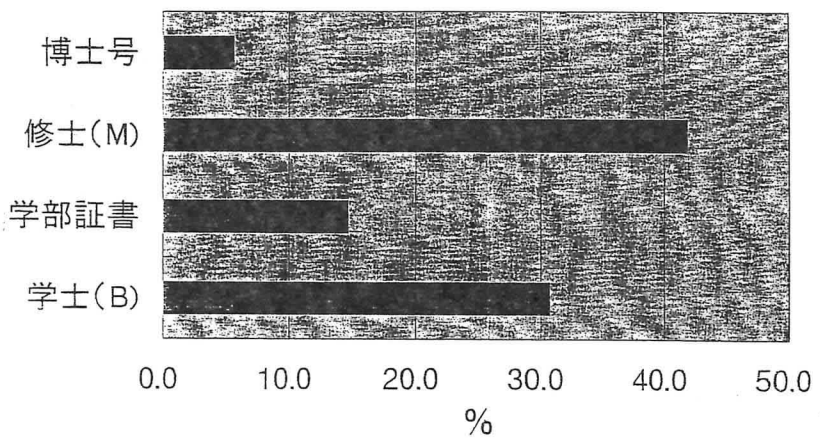


図１１ 取得できる学位 (287校)

＜基準＞
・ 大学院は，明らかにマスター (=修士：MBA，MS，....) だけカウントした．
・ マスター以外の大学院終了プログラムは，期間の短いものもあり，よくわからなかったのでカウントしなかった．
・ 学部は，バチェラー (=学士：BXX) と，それ以外の学部証書をわけてカウントした．

図１１より博士号を出す大学も 5%強 (16件) もある．注目に値するが，博士号を遠隔学習だけで完全に取得するのは難しいであろう．これは，Ph.D か Ed.Dである．修士 (41.8%) が，学士 (30.7%) より多いのは予想外だった．これは，大学院レベルになれば，レポートやディスカッションの内容のほうが評価され，試験が課されることはほとんどない (または，少ない) からと考えれば納得がいく．年月も短くて済む．修士のほうが遠隔学習に向いている市場だといえそうである．また，上位の教育ほど師弟関係で行う日本の教育と異なり，米国の場合，大学も基礎的レベルほど少人数でしっかり教えるべきだという考え方があり，その関係もあるだろう．データ数が不充分なため示していないが，現実問題として最長在籍期間も学校選びの重要なポイントである．一般に，米国より英国のほうが期間が短く，その点厳しいようである．

図１２は，大学が提供しているコースの種類の組み合わせである．これから，学部＋院 (＋無単位

コース) の総合大学が約50%，学部 (＋無単位コース)，つまり学部単位のリベラルアーツとよばれる大学が約30%，院 (＋無単位コース)，つまり研究中心の大学院大学が20%含まれることがわかる．

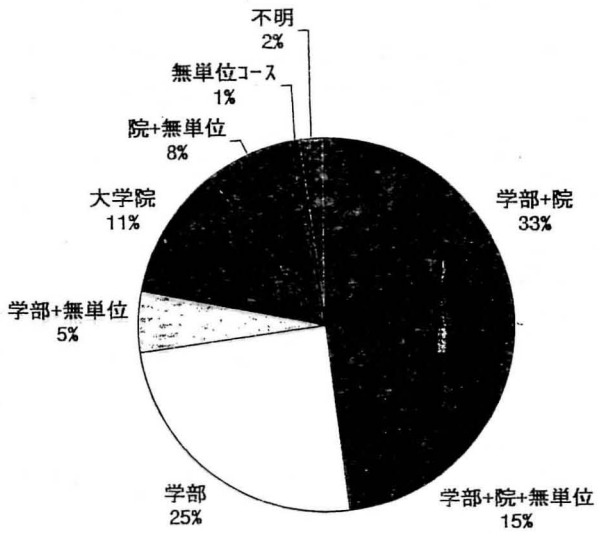


図１２ 設置コース (287校)

校で得た単位が，政府立・州立の大学では認められないことがある．コースに入学する前に，その単位が互換先の大学でも認められるか否かを確認する必要がある．遠隔学習では単位の相互互換が頻繁になってきており，無視できない問題である．

６．まとめ

以上，日本から受講できそうな，主として米国の遠隔学習大学について調べてみてわかったことをまとめると，

(1) Webを使わないところでも，電子メールはどんどん使っていること．
(2) ビデオ会議(51.2%)の普及 CATVの影響．
(3) オーディオ会議(23.7%)の普及．
(4) TVも思ったより高い利用率(～40%)．
(5) アクレディットの問題があること．
(6) 単位互換の問題があること．
(7) 通信教育も含めて，海外の大学の歴史的重み，または先進性．

もともとタイプライタを使っていた国だから，電子メールへの移行はスムーズだったはずで，(1) は誰でも予測できる．(2) に関して，インターネットと同じくらいCATVが使われている可能性があるが (米国のCATVのインフラ利用)，これは発想としてなかった．(4) は，放送大学の地域版でやはりCATVのインフラが寄与していると捉えている．アクレディテーションや単位互換の問題はあるが，全米の高等教育機関の93% が，試験によって知識と技能を確認できれば，実際の学習以前に単位を認定するといっている．

大切な問い掛け(当り前のことだが)は，

・自分の仕事や生活のパターンとマッチングするか．
・余裕があるか(時間的にも精神的にも)．
・サポート体制のしっかりしている大学を選んでいるか．

である．

今回は，SIGEDUの２月月例会の話のうち，具体的な大学名や遠隔学習のホームページの解説は割愛させて頂いた．不足部分については，稿を改めてまた書こうと考えているので，ご容赦願いたい．今回，取り上げた 287校については学校名のリストを添付した．本論が，遠隔学習や通信教育の知識

をまとめることに少しでも役立てば幸いである.

参考文献

[1] Jon Latimer, "Peterson's Guide to Distance Learning Programs 1998", Peterson's, November(NJ, 1997). 笠木 恵司 訳, 「日本で学べるアメリカの遠隔学習プログラム」, ダイヤモンド社, (1998). (原書は 800大学以上を掲載. 訳本は, このうち 446大学を掲載).
[2] "(Peterson's) Guide to Distance Learning Programs 2002", Peterson's, (NJ, 2001).
[3] 「eラーニング白書 2001/2002年度版」, 先進学習基盤協議会(ALIC), (2001).
[4] 「平成 12年版 厚生白書」, 厚生省, (2000).
[5] 「平成 11年度 我が国の文教施策 -進む教育改革-」 (教育白書), 文部省, (1999).
[6] 「世界原色百科事典」初版, 小学館, 第 8 巻(1967), p253.
[7] 久野 吉久, 「わが国の通信教育はどのように開発されたか」 (「2002各種通信教育ガイド」の解説記事), 自由国民社, (2001), p47.
[8] http://www.londonexternal.ac.uk/distancelearning/distancestudy.html "Distance Learning and the University of London".
[9] http://www.sanno.ac.jp/tukyo/inet.html 「産能大学・産能短期大学 通信教育課程 iNetCampus」.
[10] たとえば 「gain」 (産能大学通信教育課程の月次通信誌), October(2001), p23, p35.
[11] 「遠隔教育の利用動向調査 (全国)」, 先進学習基盤協議会(ALIC), (2001).

付表 抽出 287校のリスト (1/3)

表 抽出データ287校 リスト

連番	大学名	国	住所	アクレディット	設立年	通教開始年
1	Acadia Univ.	カナダ	Nova Scotia	地域	1838	1968
2	Athabasca Univ.	カナダ	Alberta	地域	1970	1972
3	Brock Univ.	カナダ	Ontario	地域	1964	1994
4	Carleton Univ.	カナダ	Ontario	州	1942	1978
5	Dalhousie Univ.	カナダ	Nova Scotia	州	1818	
6	Lakehead Univ.	カナダ	Ontario	州	1965	1987
7	Laurentian Univ.	カナダ	Ontario	州	1960	1972
8	Mount Saint Vincent Univ.	カナダ	Nova Scotia	州	1873	1980
9	Ryerson Polytechnic Univ.	カナダ	Ontario	州	1948	1970
10	Universite Laval	カナダ	Quebec	州	1852	1984
11	Univ. of British Columbia	カナダ	B. Columbia	州	1915	1949
12	Univ. of Calgary	カナダ	Alberta	州	1945	1977
13	Univ. of Guelph	カナダ	Ontario	州	1964	1984
14	Univ. of Saskatchewan	カナダ	Saskatchew	州	1907	1920
15	Wilfrid Laurier Univ.	カナダ	Ontario	州	1911	1978
16	Heriot-Watt Univ.	英国	Edinburgh		1966	1990
17	Univ. of Leicester	英国	Leicester			1991
18	Adams State College		Colo.	NC	1921	1988
19	Allegheny Univ of the Health Sciences		Pa.	MS	1848	1995
20	American College		Pa.	MS	1927	1927
21	American Military Univ.		Va.	通教	1991	1991
22	Arizona State Univ. West		Ariz.	NC	1984	1995
23	Assemblies of God Theological Seminar		Mo.	NC	1972	1980
24	Atlantic Union College		Mass.	NE	1882	1972
25	Auburn Univ.		Ala.	南部	1856	1976
26	Baker College		Mich.	NC	1911	1994
27	Bastyr Univ.		Wash.	NW	1978	1992
28	Bellevue Univ.		Neb.	NC	1965	1985
29	Bemidji State Univ.		Miss.	NC	1919	1977
30	Bethel Theologocal Seminary		Minn.	NC	1871	1994
31	Black Hills State Univ.		S. Dak.	NC	1883	1994
32	Brigham Young Univ.		Utah	NW	1875	1921
33	Caldwell College		N. J.	MS	1939	1979
34	California College for Health Sciences		Calif.	通教	1978	1978
35	California Institute of Integral Studies		Calif.	西部	1968	1992
36	California State Univ., Dominguez Hills		Calif.	西部	1960	1975
37	California State Univ., Fullerton		Calif.	西部	1957	
38	California State Univ., Hayward		Calif.	西部	1957	1993
39	California State Univ., Los Angeles		Calif.	西部	1947	1978
40	Calvin College		Mich.	NC	1876	1995
41	Central Baptist Theological Seminary		Kan.	NC	1901	1989
42	Central Michigan Univ.		Mich.	NC	1892	1993
43	Central Missouri State Univ.		Mo.	NC	1871	1993
44	Champlain College		Vt.	NE	1878	1993
45	Chapman Univ.		Calif.	西部	1861	
46	Charter Oak State College		Conn.	NE	1973	1992
47	Christopher Newport Univ.		Va.	南部	1961	1993
48	Cincinnati Bible College and Seminary		Ohio	NC	1924	1994
49	City Univ.		Wash.	NW	1973	1985
50	Cleary College		Mich.	NC	1883	1995
51	College for Financial Planning		Colo.	NC		1972
52	College of St. Scholastica		Minn.	NC	1912	1986
53	The College of West Verginia		Va.	NC	1933	1992
54	Colorado State Univ.		Colo.	NC	1870	1967
55	Columbia International Univ.		S. C.	南部	1923	1978
56	Concordia Univ.		Ill.	NC	1864	1950
57	Concordia Univ. at Austin		Tex.	南部	1926	1992
58	Concordia Univ. Wisconsin		Wis.	NC	1881	1992
59	Cornell Univ.		N. Y.	MS	1865	
60	Dakota State Univ.		S. Dak.	NC	1881	1989
61	Dallas Theologocal Seminary		Tex.	南部	1924	1987
62	David N. Myers College		Ohio	NC	1848	1995
63	Denver Conservative Baptist Seminary		Colo.	NC	1950	1988
64	Duke Univ.		N. C.	南部	1838	1996
65	Duquesne Univ.		Pa.	MS	1878	1996
66	East Carolina Univ.		N. C.	南部	1907	
67	Eastern Illinois Univ.		Ill.	NC		
68	Eastern Kentucky Univ.		Ky.	南部	1906	1941
69	Eastern Michigan Univ.		Mich.	NC	1849	
70	Eastern Oregon Univ.		Oreg.	NW	1929	1978
71	Eastern Washington Univ.		Wash.	NW	1882	1970
72	Embry-Riddle Aeronautical Univ., Exten		Fla.	南部	1970	1980
73	Emporia State Univ.		Kan.	NC	1863	1970
74	Fielding Institute		Calif.	西部	1974	1974
75	Florida Atlantic Univ.		Fla.	南部	1961	
76	Florida State Univ.		Fla.	南部	1857	1987
77	Franciscan Univ. of Steubenville		Ohio	NC	1946	1995
78	Franklin Univ.		Ohio	NC	1902	1996
79	Gannon Univ.		Pa.	MS	1925	
80	The George Washington Univ.		Wash. D. C.	MS	1821	
81	George Institute of Technology		Ga.	南部	1885	1977
82	Glenville State College		Va.	NC	1872	1980
83	GMI Engineering & Management Institut		Mich.	NC	1919	1982
84	Gordon-Conwell Theological Seminary		Mass.	NC	1884	1986
85	Governors State Univ.		Ill.	NC	1969	1981
86	Graceland College		Iowa	NC	1895	1988
87	The Graduate School of America		Minn.	NC		1993
88	Grand Rapids Baptist Seminary		Mich.	NC		1969
89	Griggs Univ.		Md.	通教		
90	Hampton Univ.		Va.	南部	1868	1983
91	Illinois Institute of Technology		Ill.	NC	1890	1976
92	Indiana Univ. System		Ind.			1921
93	Indiana Wesleyan Univ.		Ind.	NC	1920	1992
94	Institute of Transpersonal Psychology		Calif.	西部	1975	1985
95	International Univ.		Colo.	NC		1995
96	Iowa State Univ. of Science and Techn		Iwoa	NC	1858	
97	ISIM Univ.		Colo.	通教	1987	1992
98	Johnson Bible College		Tenn.	南部	1893	1988
99	Kansas State Univ.		Kan.	NC	1863	1919
100	Lehigh Univ.		Pa.	MS	1865	1992
101	Lesley College		Va.	NE	1909	1996
102	Liberty Univ.		Ill.	南部	1971	1985
103	Lincoln Christian College		Calif.	NC	1944	1993

付表 抽出 287校のリスト (2/3)

104	Loma Linda Univ.	Calif.	西部	1905	1996
105	Los Angeles Community College Distric	Calif.			1964
106	Louisiana College	La.	南部	1906	1995
107	Lycoming College	Pa.	MS	1812	1996
108	Marywood Univ.	Pa.	MS	1915	1975
109	Massachusetts Institute of Technology	Mass.	NE	1861	1996
110	McGraw-Hill World Univ.	Wash. D. C.	通教	1992	1997
111	The McGregor School of Antioch Univ.	Ohio	NC	1988	1988
112	Mercy College	N. Y.	MS	1951	1992
113	Metropolitan State College of Denver	Colo.	NC	1963	1990
114	Michigan State Univ.	Mich.	NC	1855	
115	Michigan Technological Univ.	Mich.	NC	1885	1984
116	Middle Tennessee State Univ.	Tenn.	南部	1911	1994
117	Minot State Univ.	N. Dak.	NC	1913	1991
118	Mississippi State Univ.	Miss.	南部	1878	1989
119	Montana State Univ.-Bozeman	Mont.	NW	1893	
120	Montana Tech of The Univ. of Montana	Mont.	州	1895	1980
121	Morehead State Univ.	Ky.	南部	1922	1995
122	National Technological Univ.	Colo.	NC		1984
123	National Univ.	Calif.	西部	1971	1994
124	New Hampshire College	N. H.	NE	1932	1996
125	New Jersey Institute of Technology	N. J.	MS	1881	1985
126	New School for Social Reserch	N. Y.	MS	1919	1994
127	New York Institute of Technology (NIT)	N. Y.	MS	1955	1985
128	New York Univ.	N. Y.	MS	1831	1992
129	Nicholls State Univ.	La.	南部	1948	
130	North Carolina State Univ.	N. C.	南部	1887	1976
131	North Dakota State Univ.	N. Dak.	NC	1890	1974
132	Northeastern Univ.	Mass.	NE	1898	1983
133	North Arizona Univ.	Ariz.	NC	1899	
134	Northern State Univ.	S. Dak.	NC	1901	
135	Northwestern College	Minn.	NC	1902	1994
136	Northwood Univ.	Mich.	NC	1959	
137	Nova Southeastern Univ.	Fla.	南部	1964	1984
138	Oakland Univ.	Mich.	NC	1957	1995
139	The Ohio State Univ.	Ohio	NC	1870	
140	Ohio Univ.	Ohio	NC	1804	
141	Ohio Univ.-Eastern	Ohio	NC	1957	1985
142	Oklahoma State Univ.	Okla.	NC	1890	
143	Oral Roberts Univ.	Okla.	NC	1963	1975
144	Oregon State Univ.	Oreg.	NW	1868	
145	Our Lady of the Lake Univ. of San Anto	Tex.	南部	1895	1996
146	Pacific Oaks College	Calif.	西部	1945	1996
147	Park College	Mo.	NC	1875	1996
148	Pennsylvania State Univ. Univ. Park Ca	Pa.	MS	1855	1892
149	Piedmont College	Ga.	南部	1897	1994
150	Polytechnic Univ., Brooklyn Campus	N. Y.	MS	1854	1985
151	Polytechnic Univ., Westchester Graduat	N. Y.	MS		
152	Purdue Univ.	Ind.	NC	1869	1965
153	Reformed Theological Seminary	Fla.	米加神学校連合		1991
154	Regent Univ.	Va.	南部	1977	1991
155	Regis Univ.	Colo.	NC	1877	1992
156	Rend Lake College	Ill.	NC	1967	1995
157	The Richard Stockton College of New Jer	N. J.	MS	1971	1996
158	Rochester Institute of Technology	N. Y.	MS	1829	1980
159	Roger Williams Univ.	R.I.	NE	1956	1974
160	Roosevelt Univ.	Ill.	NC	1945	1974
161	Rosemont College	Pa.	MS	1921	1996
162	Rutgers, The State Univ. of New Jersey,	N. J.	MS	1892	1996
163	Sain Francis College	Pa.	MS	1847	1994
164	Saint Joseph's College	Maine	NE	1912	1976
165	St. Thomas Aquinas College	N. Y.	MS	1952	1996
166	Salve Regina Univ.	R.I.	NE	1934	1985
167	San Diego State Univ.	Calif.	西部	1897	1984
168	Seabury-Western Theological Seminary	Ill.	NC	1933	1995
169	Seattle Pacific Univ.	Wash.	NW	1891	1984
170	Shenandoah Univ.	Va.	南部	1875	1988
171	Simpson College	Iowa	NC	1860	1996
172	Skidmore College	N. Y.	MS	1903	1971
173	Southern Methodist Univ.	Tex.	南部	1911	1962
174	Southern Polytechnic State Univ.	Ga.	南部	1948	1995
175	Southern Adventist Univ.	Tex.	南部	1894	1978
176	Southwestern Assemblies of God Univ.	Tex.	南部	1927	1983
177	Southwest Missouri State Univ.	Mo.	NC	1905	1974
178	Southwest State Univ.	Minn.	NC	1963	1968
179	Sotthwest Texas State Univ.	Tex.	南部	1899	1953
180	Spertus Institute of Jewish Studies	Ill.	NC	1925	1995
181	Stanford Univ.	Calif.	西部	1891	1969
182	State Univ. of New York at Binghamton	N. Y.	MS	1946	1989
183	State Univ. of New York at New Paltz	N. Y.	MS	1828	1995
184	State Univ. of New York College of Enviro	N. Y.	MS	1911	1996
185	State Univ. of New York Empire State Col	N. Y.	MS	1971	1982
186	Stephens College	Mo.	NC	1833	1971
187	Stevens Institute of Technology	N. J.	MS	1870	1993
188	Syracuse Univ.	N. Y.	MS	1870	1966
189	Taylor Univ., Fort Wayne Campus	Ind.	NC	1938	1992
190	Temple Baptist Seminary	Tenn.	ﾊﾞｲﾌﾞﾙｶﾚｯｼﾞ教会		1993
191	Tennessee Temple Univ.	Tenn.	ﾊﾞｲﾌﾞﾙｶﾚ	1946	1988
192	Texas Tech Univ.	Tex.	南部	1923	1927
193	Texas Wesleyan Univ.	Tex.	南部	1890	1996
194	Thomas College	Maine	NE	1894	1997
195	Thomas Edison State College	N. J.	MS	1972	1972
196	Towson Univ.	Md.	MS	1866	1995
197	Trinity College of Vermont	Vt.	NE	1925	1995
198	Trinity International Univ.	Ill.	NC	1897	1979
199	Troy State Univ.-Florida Region	Fla.	南部		1992
200	Troy State Univ. Montgomery	Ala.	南部	1957	1987
201	The Union Institute	Ohio	NC	1964	1969
202	United States Sports Academy	Ala.	南部	1972	1993
203	The Univ. of Akron	Ohio	NC	1870	1994
204	The Univ. of Alabama	Ala.	南部	1831	1991
205	The Univ. of Alabama in Huntsville	Ala.	南部	1950	1992
206	Univ. of Arizona	Ariz.	NC	1885	1972
207	Univ. of Arkansas	Ark.	NC	1871	
208	Univ. of Bridgeport	Conn.	NE	1927	1997
209	Univ. of Callfornia, Davis	Calif.	西部	1905	1987

付表 抽出 287校のリスト (3/3)

210	Univ. of California Extension	Calif.			1913
211	Univ. of Central Oklahoma	Okla.	NC	1890	
212	Univ. of Charleston	Va.	NC	1888	1982
213	Univ. of Cincinnati	Ohio	NC	1819	1976
214	Univ. of Colorado at Boulder	Colo.	NC	1876	1983
215	Univ. of Colorado at Colorado Springs	Colo.	NC	1965	
216	Univ. of Dallas	Tex.	南部	1955	1970
217	Univ. of Dayton	Ohio	NC	1850	
218	Univ. of Georgia	Ga.	南部	1785	1985
219	Univ. of Great Falls	Mont.	NW	1932	
220	Univ. of Houston	Tex.	南部	1927	1984
221	Univ. of Idaho	Idaho	NW	1889	1976
222	Univ. of Illinois at Urbana-Champaign	Ill.	NC	1867	1925
223	Univ. of Kansas	Kan.	NC	1866	1891
224	Univ. of Kentucky	Ky.	南部	1865	1974
225	Univ. of Maine	Maine	NE	1865	1989
226	Univ. of Maine at Fort Kent	Maine	NE	1878	
227	Univ. of Maryland, Baltimore	Md.	MS	1807	1992
228	Univ. of Maryland, College Park	Md.	MS	1856	
229	Univ. of Maryland Univ. College	Md.	MS	1947	1972
230	Univ. of Massashusetts Amherst	Mass.	NE	1863	1974
231	Univ. of Massashusetts Dartmouth	Mass.	NE	1895	1995
232	Univ. of Massachusetts Lowell	Mass.	NE	1894	1995
233	Univ. of Memphis	Tenn.	南部	1912	1985
234	Univ. of Michigan	Mich.	NC	1817	1969
235	Univ. of Minnesota, Twin Cities Campus	Minn.	NC	1851	1909
236	Univ. of Missouri-Columbia	Mo.	NC	1839	1911
237	Univ. of Missouri-Kansas City	Mo.	NC	1929	
238	Univ. of Missouri-St. Louis	Mo.	NC	1963	
239	Univ. of Montana-Missoula	Mont.	NW	1893	1989
240	Univ. of Nebraska-Lincoln	Neb.	NC	1869	1909
241	Univ. of Nevada, Reno	Nev.	NW	1874	
242	Univ. of New Orieans	La.	南部	1958	1985
243	Univ. of North Carolina at Chapel Hill	N. C.	南部	1789	1913
244	Univ. of Notre Dame	Ind.	NC	1842	1995
245	Univ. of Oklahoma	Okla.	NC	1890	1913
246	Univ. of Pennsylvania	Pa.	MS	1740	1994
247	Univ. of Phoenix	Ariz.	NC	1976	1989
248	Univ. of Pittsburgh	Pa.	MS	1787	1972
249	Univ. of St. Thomas	Minn.	NC	1885	1994
250	Univ. of Sarasota	Fla.	南部	1974	1993
251	Univ. of South Dakota	S. Dak.	NC	1862	1967
252	Univ. of Southern Colorado	Colo.	NC	1933	1987
253	Univ. of Southern Indiana	Ind.	NC	1965	1994
254	Univ. of Southern Mississippi	Miss.	南部	1910	1913
255	Univ. of South Florida	Fla.	南部	1956	1983
256	Univ. of St. Augustine for Health Sciences	Fla.	通教	1978	1979
257	Univ. of Tennessee, Knoxville	Tenn.	南部	1794	1923
258	Univ. of Tennessee Space Institute	Tenn.	南部		1982
259	Univ. of Texas at Arlington	Tex.	南部	1895	1973
260	Univ. of Texas at Austin	Tex.	南部	1883	1909
261	Univ. of Texas at Tyler	Tex.	南部	1971	1991
262	Univ. of the State of New York, Regents (	N. Y.	MS	1970	1970
263	Univ. of Toledo	Ohio	NC	1872	1996
264	Univ. of Utah	Utah	NW	1850	1916
265	Univ. of Vermont	Vt.	NE	1791	1990
266	Univ. of Virginia	Va.	南部	1819	1984
267	Univ. of Washington	Wash.	NW	1861	1915
268	Univ. of West Florida	Fla.	南部	1963	1988
269	Univ. of Wisconsin-Extension	Wis.			1981
270	Univ. of Wisconsin-Madison	Wis.	NC	1848	1891
271	Univ. of Wisconsin-Stevens Point	Wis.	NC	1894	1985
272	Univ. of Wisconsin-Stout	Wis.	NC	1891	
273	Univ. of Wyoming	Wyo.	NC	1886	1954
274	Univ. System of Georgia	Ga.			
275	Upper Iowa Univ.	Iowa	NC	1857	1973
276	Valley City State Univ.	N. Dak.	NC	1890	
277	Virginia Commonwealth Univ.	Va.	南部	1838	1988
278	Virginia Polytechnic Institute and State U	Va.	南部	1872	1983
279	Walden Univ.	Minn.	NC		1970
280	Washington Sate Univ.	Wash.	NW	1890	1992
281	Weber State Univ.	Utah	NW	1889	
282	Western Conservative Baptist Seminary	Oreg.	NW	1927	1981
283	Western Washington Univ.	Wash.	NW	1893	1912
284	West Virginia Graduate College	W. Va.	NC	1972	1987
285	West Virginia Wesleyan College	Va.	NC	1890	1975
286	Wheaton College	Ill.	NC	1860	1972
287	Worcester Polytechnic Institute	Mass.	NE		1979

ICU 2001年冬期 ソフトウェア開発論授業総括

新谷 勝利
(AFS)

0. はじめに

この総括は，(1) シラバス，(2) 受講者調査の２部からなり，受講者調査については，あらかじめ生徒に SEAMAIL に As-Is のかたちで掲載するといってあります．[新谷] の項は新谷のコメントです．この学校では「授業効果調査」というものが実施されていますが，この原稿に間に合わないので，新谷が実施した調査への回答のみを採録しました．ソフトウェア工学教育を大学でどのように進めるべきかについての議論の参考になれば幸いです．

1．シラバス

1.1．授業にあたっての基本的考え方

(1) 説明することを一方的に聞くのではなく，質疑応答による双方向授業にする．

(2) 宿題は基本的に毎回出され，次回の講義の最初はその質疑応答により，前回授業の理解程度を確認して次に進む．

(3) E - mailに依る質疑応答を積極的に推進する．

(4) 全員が同じように理解することは期待しないが，少なくとも理解しない学生はいないことを期待する．そのために，種々の質疑応答の形態を準備する．

1.2．講義の終了時点で受講者に理解してもらいたいこと

(1)「ソフトウェアを開発する」というのはどんなことをするのか？

(2) ソフトウェア開発に従事するにあたり，何を知っていなければならないか？

1.3．講義スケジュール

第1回 (12月 8日)

ソフトウェア開発に関連する様々な意見を述べてもらうとともに，用語，考え方，等のレベルセットを図る．

資料： 大須賀論文，情報処理V.38，N.8,pp.669 - 681

宿題： 上記の論文を読み，自分の感じるところをA4版1ページで報告すること．

第2回 (12月15日)

宿題を一人ずつ発表してもらい，ソフトウェア開発に関する諸問題の議論をする．

資料：上記の大須賀論文

宿題：M. Cusumamo著 "Microsoft Secrets" (Free Press, 1995) を読んで，不明の用語等を列記すること．

第3回 (12月22日)

宿題の結果を黒板に列挙し，いくつかの用語を説明するとともに，ソフトウェア開発を産業として成功させたマイクロソフト社の実態について説明する．

資料： 上記の Cusumamo氏の本．

宿題： なし (お正月はリラックス)

第4回 (1月12日)

Davis の資料を用いて，ソフトウェアの全過程を通して経験的に知られている考え方を説明する．

資料： A. Davis 著 " 201 Principles of Software Development" (McGrow-Hill, 1995) の Table of Contents および Preface.

宿題：B. Boehm, "Seven Basic Principles of Software Engineering," Journal of System & Software, March, '83, pp.3-24 から任意のprincipleを 1 つ選び，それについての考察を A4版1ページで報告すること.

第5回 (1月19日)

宿題を一人一人に発表してもらって，議論する.

資料：上記の Boehm論文.

宿題：なし (前回の宿題について，今日の議論を踏まえて再考する).

第6回 (1月26日)

宿題を一人一人に発表してもらって，議論する (前回の続き).

資料：上記の Boehmの本 および Davis氏の本.

宿題：F. Brooks, "No Silver Bullet," IEEE Computer, April, 任意の項目をピックアップして，それがどのように効果があるか(またはないか) についての意見を，A4版1ページにまとめる.

第7回 (2月2日)

宿題を一人一人に発表してもらって，議論する (前回の続き)

資料：上記の Boehm氏の本，Davis氏の本，および Brooks 論文. 宿題：Davis, Boehm, Brooksにおける共通点，問題意識等にについての感想あるいは意見をA4版1ページにまとめる. また， ISO/IEC TR15504-2 に目を通しておくこと.

第8回 (2月9日)

休講.

第9回 (2月16日)

Davis, Boehm, Brooksの問題意識について議論するとともに，ソフトウェア開発方法論が実際にどのように適用されるを説明する.

資料：ISO/IEC TR15504-2

宿題：CMM Technical Report の Introductionに目を通しておくこと.

第10回 (2月23日)

ISO/IEC TR 15504-2およびCMMを通してプロセスを説明する.

資料：ISO/IEC TR15504-2 および CMM TR の Introduction

宿題：なし.

第11回 (3月2日)

試験および総合的な質疑応答.

2．受講者調査

(1) ソフトウェア工学というものが一体何を目的としたものと認識されているか？

- ソフトウェア開発に関わる学問体系を整え，それを学ぶことで，誰がやっても同じものができるようにすることが一つと，ソフトウェアの生産をなるべく自動化して，無駄なく効率の高い生産をすることが一つ.
- ソフトウェア工学とは，長年のソフトウェア開発を通して培った再利用可能な手法（good practice）をまとめ，ソフトウェア開発をより合理的に行うために，還元されるものである

と捉えた．この中には，新たな開発から生まれるgood practiceの抽出や手法を規定した標準制作も存在し，これを更新することもソフトウェア工学の欠かせない側面であると考えた．

－ ソフトウェア自体は，人の行うべき作業を計算機に代行させるための代理人の役割を果たすものであります．それで，ソフトウェア工学はいかに優れていて，効率的にソフトウェアを開発する過程を研究するものだと，私は認識しています．

－ ソフトウェア開発をより効率よく行えるようにして，開発費の削減，開発期間の短縮，品質の向上を図るための学問．

[新谷]：ソフトウェア工学は発展途上の体系であり，それが幅広い範囲を含むこと，実践的なものであることは理解してもらえたと思う．1日 2.5時間，10日で合計 25時間，これを全日のクラスに換算すると 1日 5時間として5日，すなわち 1週間コースとなる．企業内でソフトウェア工学を教えた経験からすると，これはあまりに短い時間であるし，実際の経験をしていないというハンディキャップがあり，インターネットでのフォローアップも含むリモートラーニングをも導入し，継続的刺激，学習というものを考える必要があるだろう．そのためには，モジュール化したシラバス作成が必要となるだろう．

(2) リーディングアサインメントの宿題とした論文を読んで，(1) との関係で何を感じているか？

－リーディングアサインメントの論文を読んで大変よかったと思う．ソフトウェア工学は経験から抽出された抽象的な学問であるゆえ，論文というoriginalな情報媒体を読む方が伝え手の内容を歪曲した形でない状態で受入れられ，より有効であると考える．

－まず，この講義を受けるまで，ソフトウェア工学が何であるか知る由もなかった．故に，(1)で記述したものは，講義を通じて受けた説明と読んだ論文から来るものであるから，(1の最後のブレットのもの) が感じたことの記述であるともいえる．また，論文に書いてあることについて特に感銘を受けるのは，やはり開発現場で何かしら経験をした人ではないか．理論はあっても実践が伴わないとやはり理解を深めるのはある面で難しいように思う．

－論文を読んだ結果として，(1の3番目のブレットのもの)のような認識になったわけでありますが，論文を読んで感じたことは，抽象的な概念の言語化の重要性を感じました．例えば，"No silver bullet"では，ソフトウェア開発の理想的な過程を"great designers"などとはっきりと言語として表現しています．言語化することによって，他の誰もがその概念を理解することができるようになるし，実践もすることができるようになります．今までの私自身には，言語化という発想が弱かったので印象に残りました．

－ ISO のSoftware Process Assessment の標準化や CMM 統合の動きから，世界のソフトウェア業界がその開発体制の標準化に向かっていることが分かる．また．"201 Principles"や"7 Basic Principles"で示されているように，ソフトウェア開発の際に一般的に注意すべきことが徐々に確立されつつあるが，"No silver bullets"は上の流れに逆行して，ソフトウェア問題を一発で解決できるような万能な方法論はないとしている．私も，いくら全世界標準の方法論を確立しようとも，それに従いさえすれば，世界中のどこであっても同じ品質のソフトウェアができるとは認識しておらず，私のこの認識は，日本のソフトウェア問題を指摘した大須賀論文によってサポートされる．すなわち，ソフトウェアを開発するのは人間なのだから，その人間の文化的，社会的背景が大きくソフトウェアの開発方法や理念に影響を与えるのであり，このことを考慮すれば，標準化されたソフトウェア工学は，それぞれの地域や社会に合わせたいくらかのカストマイズが必要だと思われる．

[新谷]：毎回宿題としてリーディングアサインメントを出すとしても，限定された回数で何を読んでもらいたいかには，ずいぶん悩んだ．主張に賛成するかどうかは別として，今後の勉強のために入り口になるようなものを選んだつもりである．(1) で述べた観点から，モジュラー化した分野を代表する論文，前面に渡るもの等に分類したほうがよかったと思っている

が，今回は幅の広さを主たる選択理由にした．

(3) 今回の授業から今後どの分野のデシプリンを学習したいと思ったか？

- 私の中では，企業というものは完成されたものしか市場にださないという偏見が存在したので"good enough"の考え方がショックでした．確かに100％の完成度を目指すことは，経済的に無駄なことかもしれない．だが，"good enough"の線引きをあまりにも低いレベルに設定してしまうのも経済的ではなく企業倫理に反するのではないか．ということで，"good enough"と企業の関係について，企業倫理などとの比較検討などを調べてみたい．もう一つは，顧客の要求を仕様化する過程，もしくは市場に存在する要求を仕様化する過程について，学習してみたい．人の思考回路をソフトウェアに託すのだから，私はここが一番ソフトウェアの難しくて重要なパートだと思っている．

- やはり基本的なことというか "201 Principles of Software Development" を始めとするソフトウェア開発の基礎理論をきちんと学んでおきたいと，今回の講義を受けて強く感じた．個人的に理系のどの分野においても，基礎がとても重要だと思うのと，基礎をおろそかにした教科が苦手科目になっているからだ (できないというより修練が足りないために応用が利かないことが多いだけなのだが).

- 外部から依頼されたソフトウェアは，どの時点になったら完成とする (クライアントに正式に納品できるレベルに達する) のかの基準について，さらに深く学習したいと思った．そのためには，最初の要求定義と仕様設計の段階が重要になるので，それらも合わせて学習してみたい．

  - ソフトウェア工学はソフトウェア開発におけるgood practiceを抽出し，体系化した学問であるが，good practiceを抽出するには，道具として他の学問領域の知識が不可欠である．今後はソフトウェア工学に関わりの強い商学や経営学などのデシプリンを学習したい．ソフトウェア工学をはじめ，自然現象を捉える学問ではなく，手法の集合を体系化した学問についても，もっと目を向けたい．

  [新谷]：2年に渡る授業，リモートラーニングによる授業，等を通してより広い範囲を，かつ，演習も含みたいと考えている．要求仕様をどうまとめるか？ 設計の文書化レベルのあり方等，設計の演習，テスト計画の考え方，等ソフトウェア開発ライフサイクルに沿い，プロセスの目標に沿い，全体的に鳥瞰するものを考えてみたい． 201 Principles は，この流れをまとめるには非常に参考になる．よって，別のシラバスとしては，201 Principles に沿ったクラスというものが考えられる．どちらのアプローチをとるにしろ，あたかもソフトウェア開発のプロジェクトブックを作成するように，システム図，モジュール構造図，モジュール間関係，等を作成すること，即ち，「ソフトウェア工学を理解する」を最終目標として，要求分析，システム設計，クラスの実装設計等がなされることになる．システム環境も定義しなければならない．ここでは，システムという用語およびその考え方が説明される必要がある．

(4) 今回どの分野，考え方をさらに議論したかったか？

- A. Davisによる 201 Principles of Software Development の内容にもっと深く入り込んで，ソフトウェア開発における基礎知識というものを，議論等を通じて理解を深めたかった．講義の後半の方法論に入る前にもっとやっておきたかった．時間的制約がきつのが残念．

- ソフトウェア工学が現代の日本のソフトウェア業界（ないしは他の業界も含む）へどのように貢献できるかについて，もう少し議論したかった．日本版 CMM に表されるように，日本のソフトウェア業界がソフトウェア工学のプリンシプルを応用できることがわかったが，その他の可能性も含めて，議論したかった．

- 何らかの形でソフトウェア工学が確立されたとして，それを実際の業務の中で効率的に有意

義に活用・適用できるソフトウェア開発者の技術的・人間的スキルとしてどのようなものがあるのか，という話題を議論してみたかった．

－ 実際に開発されたソフトウェアの開発プロセスを資料として，そのプロセスについて Validation や Verification がちゃんと実現できているかなど色々と検討してみたかった．あとは，オープンソースについて議論してみたかった．具体的には Linux と Windows のどちらが将来伸びていくか，というテーマでディベートをしてみたかったです．標準化・特許・testingなどの問題にも絡みそうで面白いと思うのですが．

[新谷]：ほんとうはa「．．をさらにクラスで勉強したかったが，それにどう自分でアプローチすればいいのかが分かったので，あとは自分で勉強しよう」という答もあるかな，と期待していたが，....

ソフトウェア工学の周辺としての，テクノロジートランスファー，知的所有権，チームダイナミックス，等をクラス途中でリーディングアサインメントでもいいから考えることができるとよかった．

(5) 次年度に受講する生徒のためにどのような改善があるかよいと思うか？

－ 今年度のように少人数であれば，今のスタイルでも問題はないと思う．積極的に質問を出来ない人に対して，問いかけをしたりして皆が参加できるスタイルになるとよりよくなりそう．時間的にも厳しい面もあると思うが．

－ 私個人の能力のためかもしれませんが，学期の最初の方はクラス目的が見えにくかったです．そのためクラスが開講する前に，薄い本でも課題図書で読んでから受講すると，クラス目的のみならず専門用語などの予習になり，短い学期を有効活用できるのではないでしょうか．あと，実際にソフトウェア開発体験をしても面白いのではないでしょうか．

－ 実際に簡単なソフトウェアを作る実習の時間を設けて，講義でやった内容を実体験できるとよいと思う．

－ 週2時間という短い講義であり多くを取り入れるのは難しいが，概念提唱の範囲から逸脱していない．ソフトウェア工学の概念を提えるにあたって，具体例や実体験が必要不可欠である．先生は概念を説明するにあたって，様々な例を紹介されたが，ソフトウェア工学を用いた総合演習があってもよいのではないかと思う．

[新谷]：ここで述べられらていることにどのように対応するかについては，既に１から３への追記で述べた．土曜日の1時半から4時までという時間帯に全員が最初から最後まで出席し，リーディングアサインメント，レポートもしてきており，質疑応答は自分からの質問は必ずしも多くはないが，こちらからの問いかけには応答する熱心な学生がいてこそこのクラスは所期の目的をまずは達成できたと考える．前述のコメントに見られるように，これは入口でしかない．

Call for Papers

ISFST2002

International Symposium on Future Software Technology 2002

October 20-21, 2002 in Xi'an (Pre-Symposium Tutorials)
October 23-25, 2002 in Wuhan (Symposium)

Co-sponsored by

Software Engineers Association (Japan)
UNU International Institute for Software Technology (Macau)
Northwest University (Xi'an, China)
and
Huazhong University of Science and Technology (Wuhan, China)

ISFST2002 is the 7th International Symposium on Future Software Technology.

The previous six symposia in this series were successfully held in China since 1996. The series have been co-organized by SEA (Software Engineers Association) of Japan, UNU/IIST (International Institute of Software Technology of United Nations University), and the local host academic organizations.

This year, Northwest University (NWU) in Xi'an and Huazhong University of Science and Technology (HUST) in Wuhan will be our host.

The purpose of the Symposium is to review the recent progresses in software technology, to exchange various technical and business information, and to promote the sound industry-academia cooperation for future. It will provide a unique opportunity for practitioners, researchers and educators in software community to gather latest information and discuss the advances and the trends in the field. Program committee is looking forward to receive your submission of research papers, experience reports, and tool demonstrations related to the following topics.

Topics:

Suggested topics include, but are not limited to:

- Computer Human Interaction
- Data Mining and Knowledge Discivery
- Distributed Computing
- Formal Methods
- Internet Security
- IT Industry Experiences
- Knowledge and Data Engineering
- Mobile and Multi Agent Systems
- New Application Technologies
- New Application Domain (e.g.: eCommerce)
- Product Line Engineering
- Process and Project Management
- Requirement Engineering
- Software Development Paradigm (e.g.: Open Source)
- System Evolution and Maintenance
- Testing and Quality Management
- Web-based Information Systems

The unique feature of the ISFST symposia is that the program includes informal but intensive working group style discussion sessions besides ordinary-style paper presentations. Industry people are encouraged to propose demonstration of tools instead of submitting regular papers.

Also, this year's ISFST will feature pre-symposium tutorials. Those tutorials will be held in Xi'an, and the symposium will be held in Wuhan. Proposals for various topics are welcome.

Both cities have unique historical background and culture. So, the attendees will be able to enjoy different atmosphere of these two cities.

Paper Submission:

Submissions for a full paper or an extended abstract should be received no later than end of June, 2002 by one of the Program Co-Chairs:

Paper Submission Addresses:

(Within China)	(else)
Preparation Committee of ISFST'2002	Mr. Kazunori Shioya
Science and Technology Association in HUST	SRA-KTL, Japan
Email : hxwu@public.wh.hb.cn	Email : m2shioya@yahoo.com

Paper submissions should contain the type of the submission (Full [5000 words] or extended-abstract [1500 words]), title, author names and their affiliation and address, abstract, list of keywords, followed by the text.

Tool demo submission should contain the name of the tool, author names and their affiliation and addresses, list of keywords, followed by the text describe tool function.

Tutorial proposal should contain name and contact information of the proposer, the title of tutorial, purpose and scope, duration (full- or half-day), and summary of topics to be covered.

Only electronic submission in PDF, Portable Postscript, or Word format via e-mail will be accepted.

Important Dates:

Submission deadline:	June 30, 2002
Notification of acceptance:	August 15, 2002
Final paper due:	September 30, 2002

Paper selection will be based on originality and contribution to the topics. SEA, Japan will publish the symposium proceedings with ISBN number after the symposium by compiling the papers accepted and presented at the symposium, and the summary report of working group discussions, etc.

ISFST2002 Staffs

Honorary Co-Chairs
Huowang Chen (National Univ of Defense Technology,China)
Toshiharu Yamasaki (Former Chairman of SEA, Japan)

Organizational Co-Chairs:
Hiroyuki Fukase (Chairman of SEA, Japan)
Kegang Hao (Northwest Univ, China)
Guohui Li (HUST, China)

General Co-Chairs:
Akira Kumagai (PFU, Japan)
Yunsheng Liu(HUST, China).

Program Co-Chairs:
Kazunori Shioya (SRA-KTL, Japan)
Soo-Yong Park (Sogang Univ, Korea)
Yanxiang He (Wuhan Univ,China)

Program Committee Members:

Keijiro Araki (Kyushu Univ, Japan)
Doo-Hwan Bae (KAIST, Korea)
Doo-Kwon Baik (Korea Univ, Korea)
Daoxu Chen (Nanjing Univ, China)
Zhiguo Chen (Henan Univ, China)
Guozhong Dai (IS-CAS, China)
Shihai Dong (Peking Univ, China)
Ming Fan (ZhengZhou Univ, China)
Hyuk-Soo Han (SangMyung Univ, Korea)
Jinpeng Huai (BUAA, China)
Chris George (UNU/IIST, Macau)
Yuqing Gu (ASTI Beijing, China)
Jiang Guo (California State Univ, USA)
Hoh Peter In (Texas A&M Univ, USA)
Masao Ito (Nil Software, Japan)
Dehua Ju (ASTI Shanghai, China)
Kyo C. Kang (POSTECH, Korea)
Kiwamu Kase (RIKEN, japan)
Gargi Keeni (TCS, India)
Jung-Ah Kim (KwanDong Univ, Korea)
Chang-Hwa Kim (KangNung Univ, Korea)
Yue-Sun Kuo (Academia Sinica, Taiwan)
Kei Kurakawa (NAIST, Japan)
Danning Li (Guizhou Academy, China)
Yoshitaka Matsumura (SRA-KTL, Japan)
Deependra Moitra (Lucent Technologies, India)
Hideo Nakano (Osaka City Univ, Japan)
Takamasa Nara (Hitachi S&S, Japan)
Toshitsugu Nomura (JIPS, Japan)
Yong-Bae Oh (SuWon Univ, Korea)
Soo-Hyun Park (KookMin Univ, Korea)
Wanchai Rivepiboon (Chulalongkorn Univ, Thailand)
Beijun Shen (ECUST, China)
Tetsuo Tamai (Univ of Tokyo, Japan)
Jiro Tanaka (Univ of Tsukuba, Japan)
Hae-Sool Yang (HoSeo Univ, Korea)
Chenqing Ye (Zhejiang Univ, China)
Jinyuan You (Shanghai Jiao Tong Univ, China)
Mingyi Zhang (Guizhou Academy, China)
Songmao Zhang (IM-CAS, China)
Zhizhuo Zhao (Xiamen Univ, China)
Xichang Zhong (SEC-CAS, China)
Youliang Zhong (Hypersoft, Australia)
Zhiying Zhou (Tsinghua Univ, China)

Symposium Secretary:
Rongrong Li(HUST, China)
Kouichi Kishida (SEA, Japan)