

組織パターンの紹介

アジャイルなソフトウェア開発を行う組織のパターン

SIGSE パターンWG 沖田直幸
横河電機(株)



組織パターンとは

- 第4世代のソフトウェア開発のためのパターンランゲージ
 - アジャイルなソフトウェア開発を行うための組織を育てるためのパターンランゲージ
- アジャイル
 - 第4世代のソフトウェア開発の特徴の1つ
- 組織
 - ソフトウェアを開発する土壌



第4世代のソフトウェア開発

- 第1世代: コンソールスイッチ
- 第2世代: 対話的プログラミング
 - 少人数の開発、スーパープログラマ
- 第3世代: 工業的ソフトウェア開発
 - H/W, 工業的手法の導入
 - 階層的な組織構造と定型的なプロセス
- 第4世代: 動的なソフトウェア開発
 - フラットで動的な組織構造とコミュニケーションパス
 - 付加価値に直結した活動
 - アジャイルもその一面



組織への取り組み

- プロセスの限界
 - '90年代のプロセス主導開発の結果を調査
 - プロセスが答えでなければ何？
- 組織の構造に着目
 - 役割間の関係
 - 優れた生産性を持つ組織構造の共通性
 - 本当のプロセスが生まれる場所
 - 組織を癒し、補修し、育てる



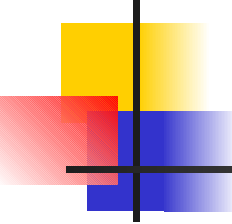
組織パターンの歴史

- '90 前半 ~ 現在 実際の開発組織の事例研究
 - Borland QPW, AT&T...
 - 100 以上の組織、数 10 の会社、数ヶ国
- '95 生成的組織パターン (PLoPD)
 - Scrum への影響
 - Kent Beck がシェファード
- '04? Organizational Patterns of Agile Software Development
 - A.Cockburn, W.Cunningham, S.Berczuk らのパターンの合流
 - 価値のあるソフトウェアを効率的に生み出しつづける
 - Effective (Agile というよりも…)
例) AT&T 5ESS アジャイルではないが、高信頼な製品



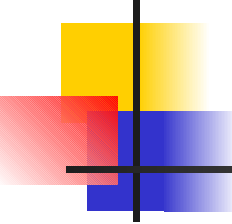
組織パターン概要

- 社会システムとしての組織
 - 組織は社会システム、形を持っている
 - 形は文化の一部、組織の価値観の表れ
- パターンランゲージ
 - 4つの視点から組織を捉える。
 - シーケンスの必要性。



組織パターンの使い方

- Step1 読んで理解する
- Step2 適用する
 - 組織の変化は文化の変化
 - 簡単ではない。危険でもある。
 - シーケンスの必要性 (ランゲージ)
 - 人の関わり。会話と開発者の自発的参加
- Step3 更新する
 - 組織に合わせたテーラリング
 - パターンの追加、組織パターンへのフィードバック



4つのパターンランゲージ

- 関連しあった4つのパターンランゲージ
- 設計パターン (Design Patterns)
 - Project Management Pattern Language
 - Piecemeal Growth Pattern Language
- 構築パターン (Construction Patterns)
 - Organizational Style Pattern Language
 - People and Code Pattern Language



Organization Design Patterns

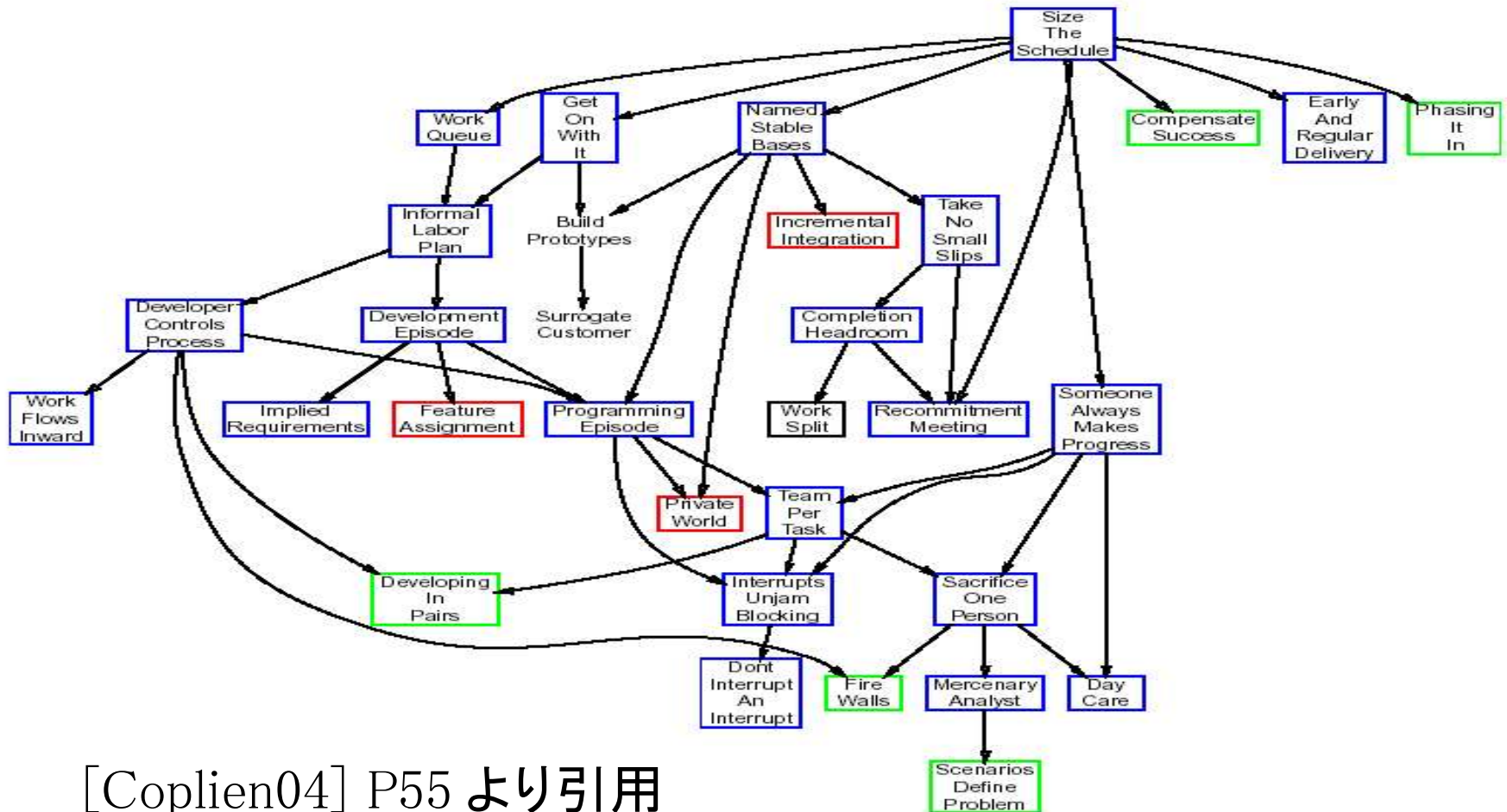
- 組織を設計するためのパターンたち
 - GoF の Design Pattern とは違う意味
 - 役割の主な関連の理解、組織の骨格形成。
 - ソフトウェアアーキテクチャに相当
 - Construction Patterns との区分ははっきりとはしていない。むしろ相互に関連。
 - 組織は成長する。組み立てられない。ToBe の姿。



Project Management Pattern Language

- プロジェクトマネジメントからの視点
- 組織が行うべき作業に着目
 - スケジュール
 - プロセス
 - タスク
 - 進捗を進めるための取り組み
- 26のパターン

Project Management P.L. (2)



[Coplien04] P55 より引用

March, 2004

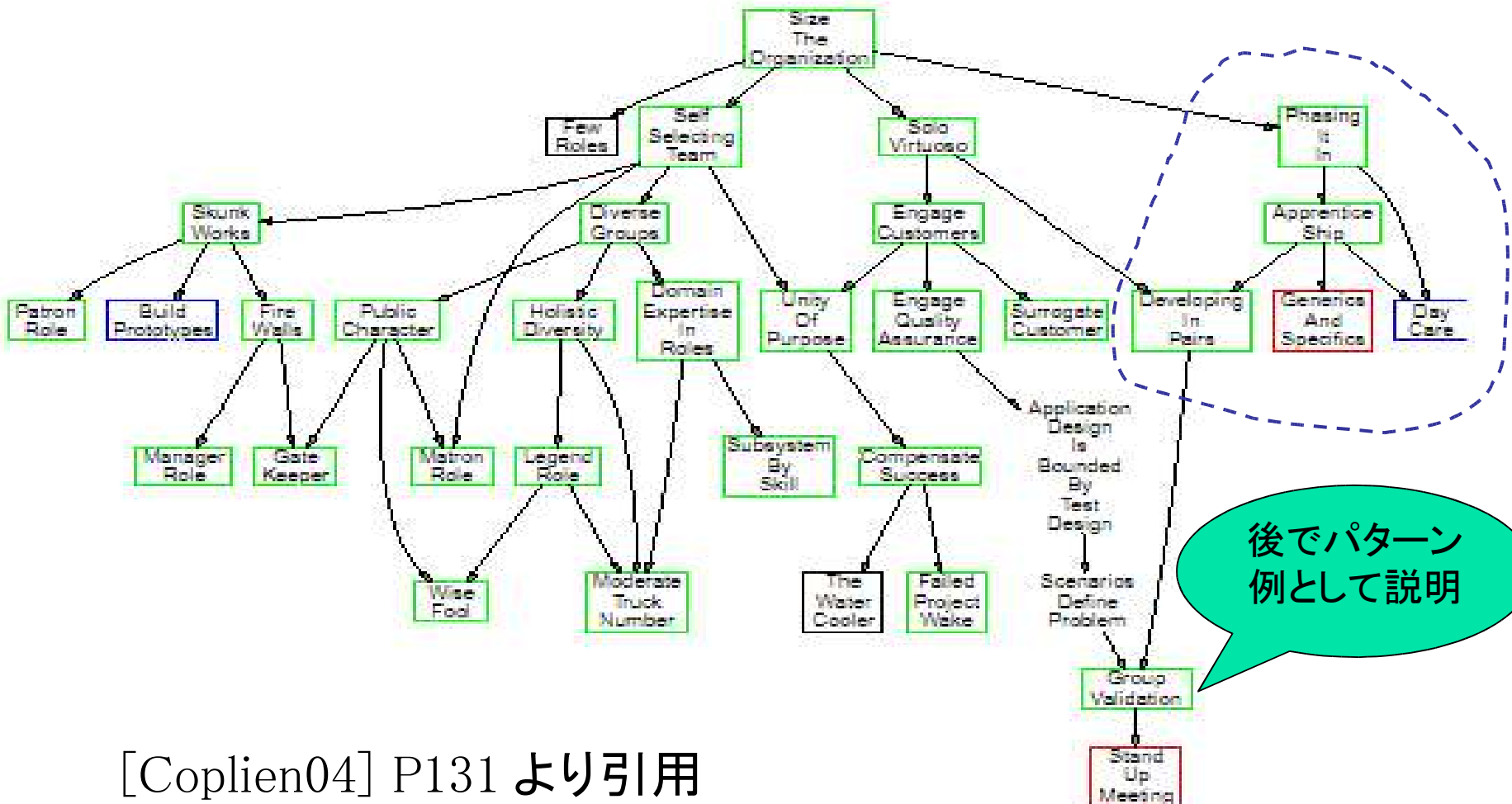
SIGSE Pattern WG N.Okita



Piecemeal Growth Pattern Language

- 組織とプロセスを共に漸進的に成長させる
 - プロセスと製品を共に成長する同時エンジニアリング
 - フィードバックと洞察を活用して、組織を強くし、チューニングする
 - 組織の修復プロセス
- 32 のパターン

Piecemeal Growth P.L. (2)



後でパターン例として説明

[Coplien04] P131 より引用



Organization Construction Patterns

- 組織を構築するためのパターンランゲージ
- コーンウェイの法則への対応
 - 組織がアーキテクチャに従うか、アーキテクチャが組織に従う。
- 日々の現実への対処。
- 漸進的に進める組織づくり



Organizational Style Pattern Language

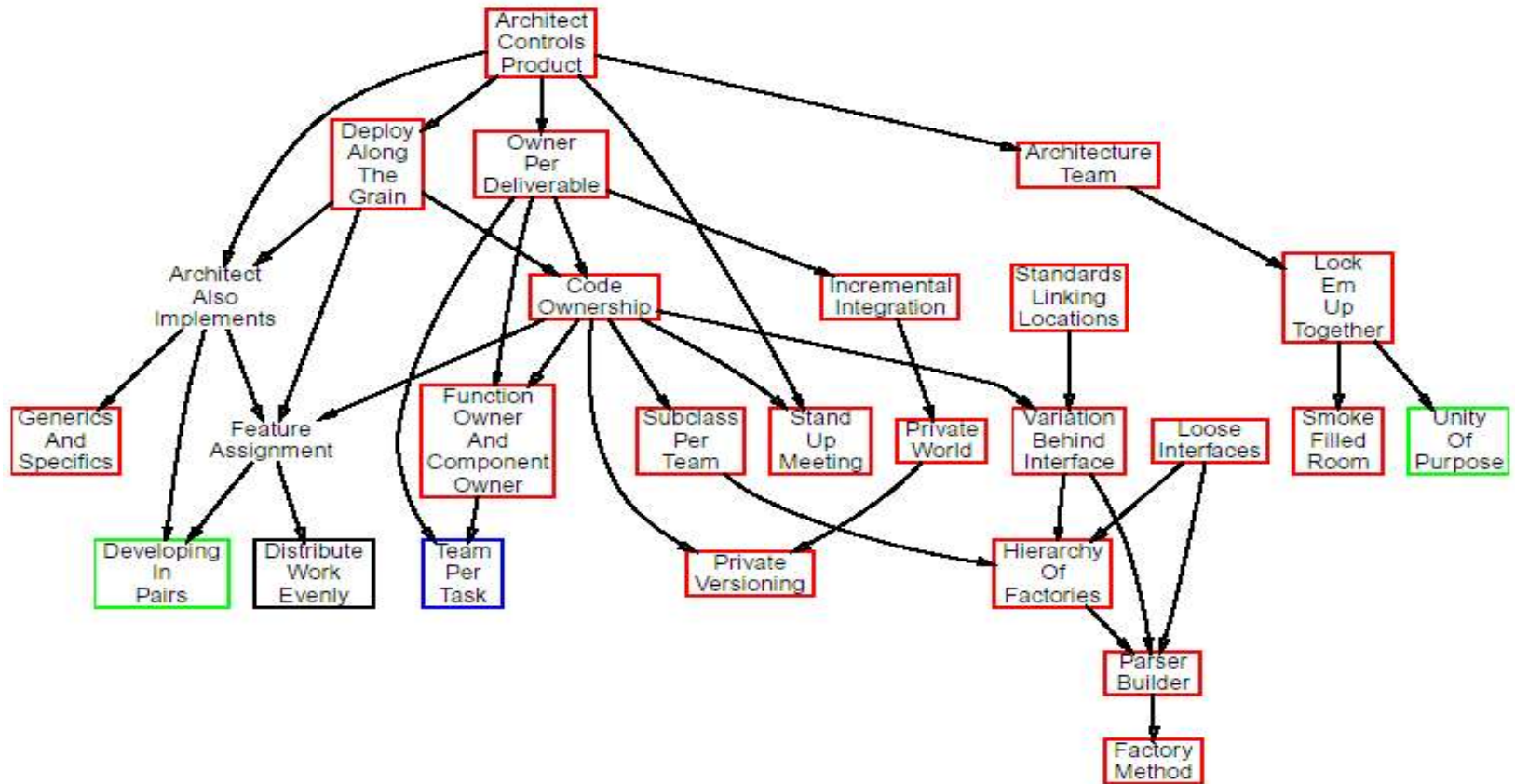
- 組織の役割間の関係に着目
 - 異なる組織にどのように展開するか
 - マネジメントスタイルと強い相関
- 23 のパターン



People and Code Pattern Language

- コンウェイの法則との整合
- 人とコードの密接な関わりに着目
 - 組織構造と製品アーキテクチャの相関
 - 役割構造 (アーキテクトと開発者の関わり)
 - 地理的構成
 - 開発スタイル (ペア開発、スタンドアアップミーティング)
- 23 のパターン

People and Code P.L. (2)



[Coplien04] P294 より引用

March, 2004

SIGSE Pattern WG N.Okita

18



パターンの形式

- Alexander 形式を意識
 - パターン名
 - 写真
 - コンテキスト (文脈)
 - 問題
 - 解法
 - 根拠
 - 事例、引用
 - 関連するパターン

門番 (Gate Keeper) パターン

門番 (GateKeeper) **



... 開発者の組織は形成され、同僚や資金提供者や顧客、その他の“部外者”に詮索されている。

プロジェクトは、交流しているまたは交流すべき多くの部外者とのよいインタフェースを必要とする。

多くのプロフェッショナルな開発者(特にプログラマ)は、ソフトウェアと関わったり技術的な仕事をする事の方が人間関係の仕事よりも心地よい。それでも、孤立主義はうまく行かない。情報の流れは重要である。一方で、コミュニケーションはコストがかかる。コミュニケーションのオーバーヘッドは外部の交流する人数に非線形に増加する。もし、多くの交流による割り込みがノイズでなければ、これはそんなに悪くない。

外部との交流が組織をコントロールしてしまうよりも、組織が外部との交流をコントロールすべきである。これは組織の成熟度の大きな指標である。

パターン名

写真

文脈

問題



門番 (Gate Keeper) パターン

Therefore:

プロジェクトメンバのうち、人を引きつけれる周知の人
(PublicCharacter) を門番の役割に昇格させなさい。この人物はプロジェクト外の最新でちょっとした情報をプロジェクトメンバに広め、それをプロジェクトに関連した用語に“翻訳”する。門番はまた、プロジェクトの情報を関連する外部の人に「漏らす」かもしれない。

* * *

この役割はまた、開発と営業 / 本社機構とのインタフェースを管理する。

このパターンは防火壁 (FireWalls) パターンとのバランスをもたらす。顧客がまだ部外者とみなせるまでは顧客を引き込め (EngageCustomers) パターンを補足する。

門番と防火壁だけでは、マーケティングが開発スケジュールの決定を許すような企業文化にいる開発者を守るには不十分である。

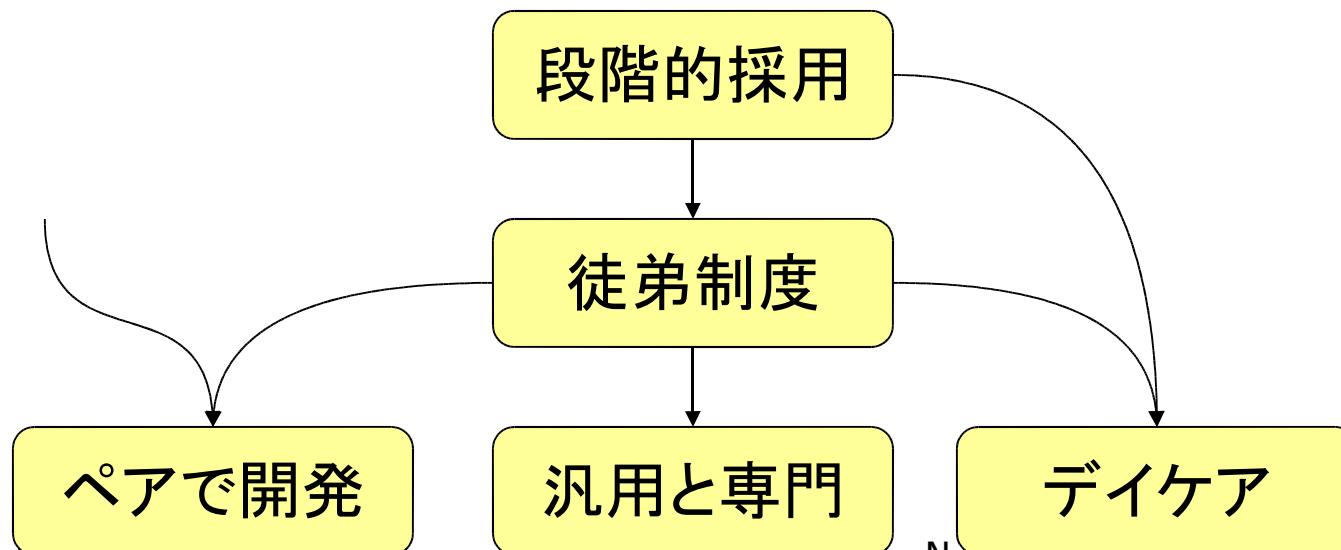
..... 以下略

解法

補足
関連パターン

パターンランゲージ実例

- パターンランゲージ
 - 問題の解決が新しい文脈を作る
 - Piecemeal Growth Pattern Language の一部





Phasing It In Pattern

段階的採用 **

... 重要なプロジェクトメンバは雇われており、必要な専門知識をカバーしている。しかし、プロジェクトはより多くのスタッフを必要としている。

プロジェクトを育てるためには、長期的にスタッフを育てる方法を明確にする必要がある。いつ、誰を何人雇うか？プロジェクトは成長の痛みを最小限にしつつ、大きくならなければなりません。 <略>

重要な人々を最初に雇うことは重要である。早い時期に重要なメンバで臨界量に達する必要がある。しかし、早期にあまりにも多いメンバは、核となるチームにとって負担となる。

Therefore:

徐々に雇うように計画しよう。最初にエキスパートを雇う、そしてプロジェクトが大きくなる必要に応じて、段階的に新しい要員を雇う。

開発の負荷に一致するまで、組織は要員を増やすことができる。このパターンは、密接に「徒弟制度」や「トラックナンバーと調整する」と関係がある。「デイケア」は、新人が組織に配属される時の、トレーニングおよび指導者の負荷を手助けするために適用することができます。



Apprentice Ship Pattern

徒弟制度 *

... プロジェクトは、最初にエキスパートを配置した後に、徐々に充足している。

* * *

プロジェクトは、プロジェクトの成長要求と高度なドメインの専門知識の発展と維持のバランスを取らなければならない。臨界に達するには、十分な要員が必要である。しかしながら、スタッフはプラグ・コンパチブルでも交換可能でもない。また、新しい作業への十分な仕事の適切な準備である、学術的なトレーニングや経験は、めったにない。

Therefore:

徒弟制度によって、新人をエキスパートに変えよう。すべての新人はエキスパートの見習いとして働くべきだ。ほとんどの見習いプログラムは、6 か月から 1 年 (パラダイムシフト を作るために必要な時間) 続くだろう。



Day Care Pattern

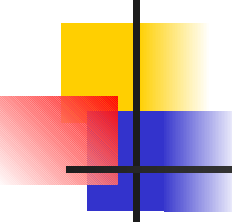
デイケア * 別名:進捗チーム / トレーニングチーム
… プロジェクトは新人を迎えた。

* * *

熟練者が新人のメンタリングにほとんどの時間を費やしている。
… 略… 新人は熟練者によってトレーニングされなければならない。
しかし同時に、プロジェクト自身も進める必要がある。

Therefore:

1 人の熟練者をすべての新人の教育に当て、残りの熟練者はシステム開発に向けよう。熟練者だけの“進捗”チームとトレーニングチームを分け、進捗チームが85-95%のシステム設計を行い、残りを品質に焦点を合わせて、トレーニングチームにさせよう。新人が役に立つぐらいに育ったら、進捗チームに移そう。



組織パターン翻訳 PJT

- 組織パターンの翻訳と勉強会
 - パターンに興味があるが英語は …
 - 有志による活動
 - 作者の J.Coplien と N.Harrison さんの協力
 - <http://www.bm-study.org/OrgPatterns/wiki/wiki.cgi?BookOutline>



参考文献

- [Coplien95] James O. Coplien : A Generative Development-Process Pattern Language
- [Coplien04] James O. Coplien, Neil Harrison : Organizational Patterns of Agile Software Development
<http://www.easycomp.org/cgi-bin/OrgPatterns?BookOutline>
- [PLoPD] James O. Coplien, Douglas C. Schmidt (eds.) Pattern Languages of Program Design
- [細谷 01] 細谷竜一, 中山裕子監訳 ソフトバンクパブリッシング : プログラムデザインのためのパターン言語 (16 章に [Coplien95] の和訳あり)