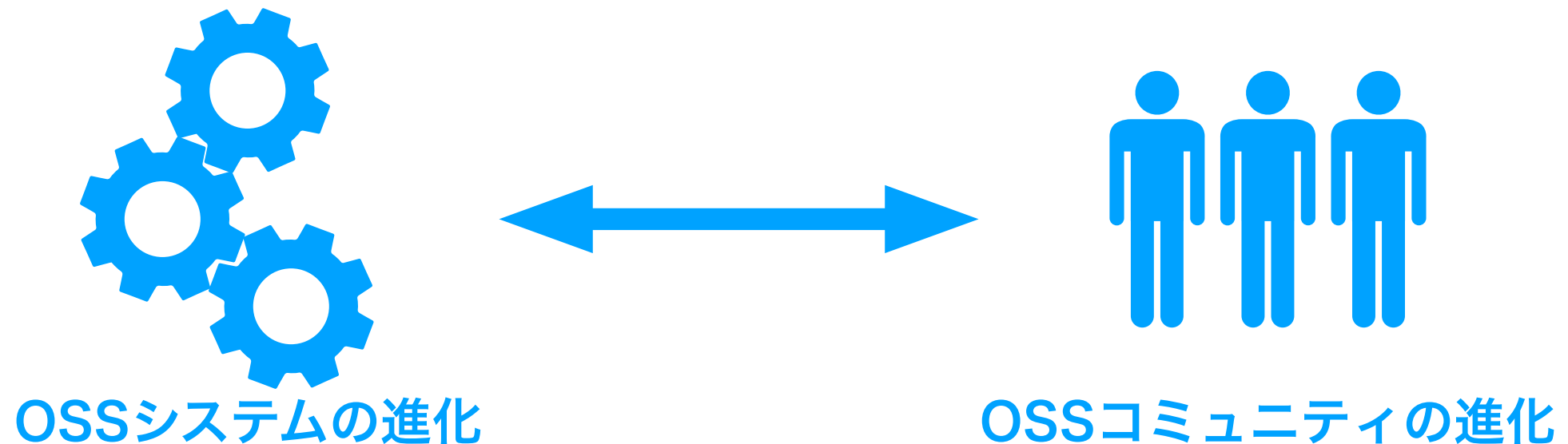


Evolution Patterns of Open-Source Software Systems and Communities

Kumiyo Nakakoji
Yasuhiro Yamamoto
Yoshiyuki Nishihara
Kouichi Kishida
Yunwen Ye

概要

- OSS開発は、自然物の進化により成功したモデルとみなされる
- OSSの進化の研究は、そのシステム自身に焦点を当てられてきた
- この論文では、より広い視野を得るためにOSSシステムの進化だけでなくOSSコミュニティの進化とそれらの進化の関係について調査した



General Framework

- OSSプロジェクトの進化の過程を分析するための全体的な骨組み
 - **Roles of Community Members**
 - ▶ コミュニティ内のメンバーの役割
 - **Community Structure**
 - ▶ コミュニティの構成
- 調査方法
 - 各OSSプロジェクトの開発者へのインタビュー
 - メーリングリストの記録の調査

Roles of Community Members

- Passive User
 - システムを利用するだけのユーザ
- Reader
 - システムを利用するだけでなくそのコードを読んで理解しようとする（レビューのような役割を果たす）
- Bug Reporter
 - バグを発見し報告する（修正はしない、テスターと同じ役割を果たす）
- Bug Fixer
 - 見つかったバグを修正する（バグが起こったコードの一部分を読む）
- Peripheral Developer
 - 新たな機能性や特徴に貢献する（稀有・不定期・短期間に）
- Active Developer
 - 新たな機能性やバグ修正に定期的に貢献する（OSSプロジェクトの主力）
- Core Member
 - OSSプロジェクトの指導や調整に責任を持っている（長期間、開発やプロジェクトの進化に大きな貢献をしてきた、Maintainersとも呼ばれる）
- Project Leader
 - 主にOSSプロジェクトの創始者（プロジェクトの方向性を決める）

Community Structure

- コミュニティ内での役割はその人物の貢献度や内容によってのみ決まる
- OSSコミュニティのメンバーのほとんどがPassive Userと言われている
 - ほとんどのOSSは少数の開発者のみによって寄与されている
- Passive UserはActive Memberを引き付ける
 - 存在自体が他のメンバーを動機づける

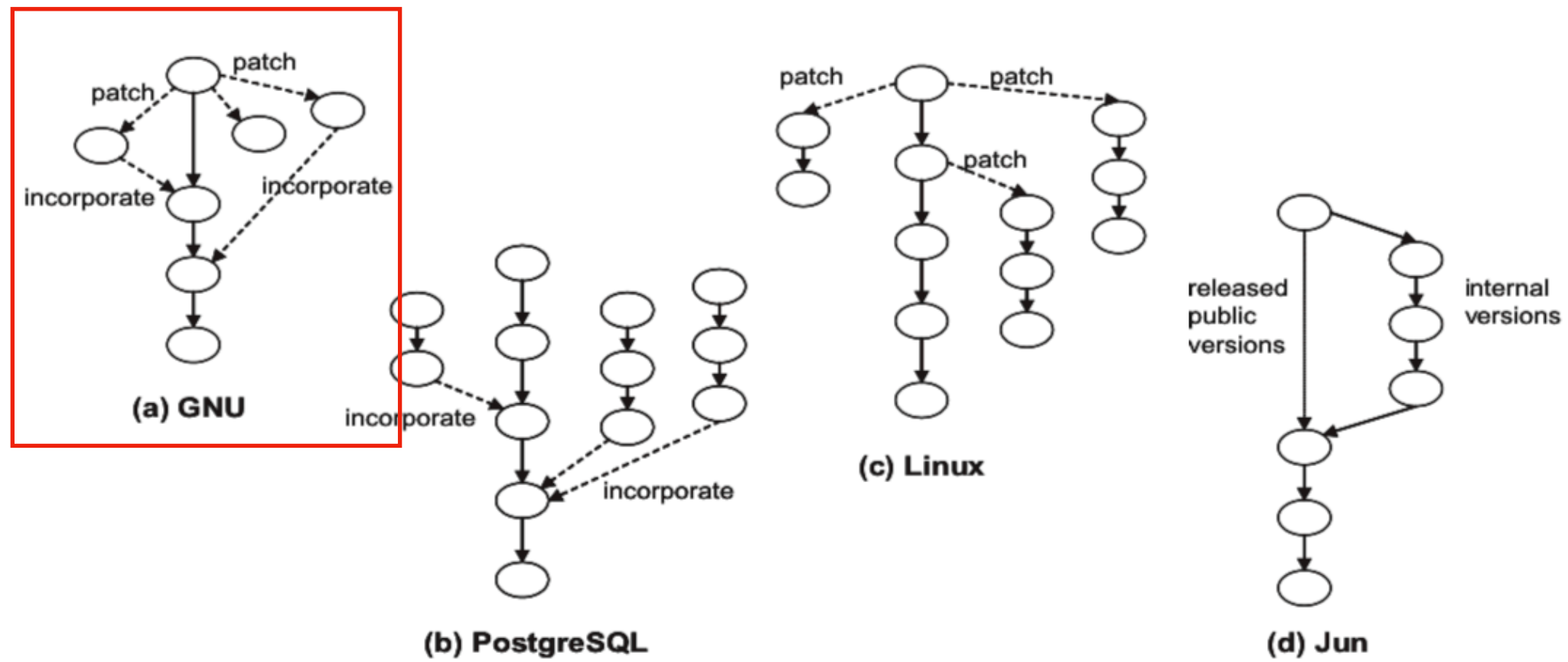
Figure 1: General structure of a F/OSS community.



1. GNUWingnut Project

- GCCやEmacsといった多くのGNUソフトウェアアプリケーションのサポートを日本の企業に提供する
- 調整のためにシンプルなコードを目指すため、開発されたパッチはコアバージョンに統合された時のみ分配される

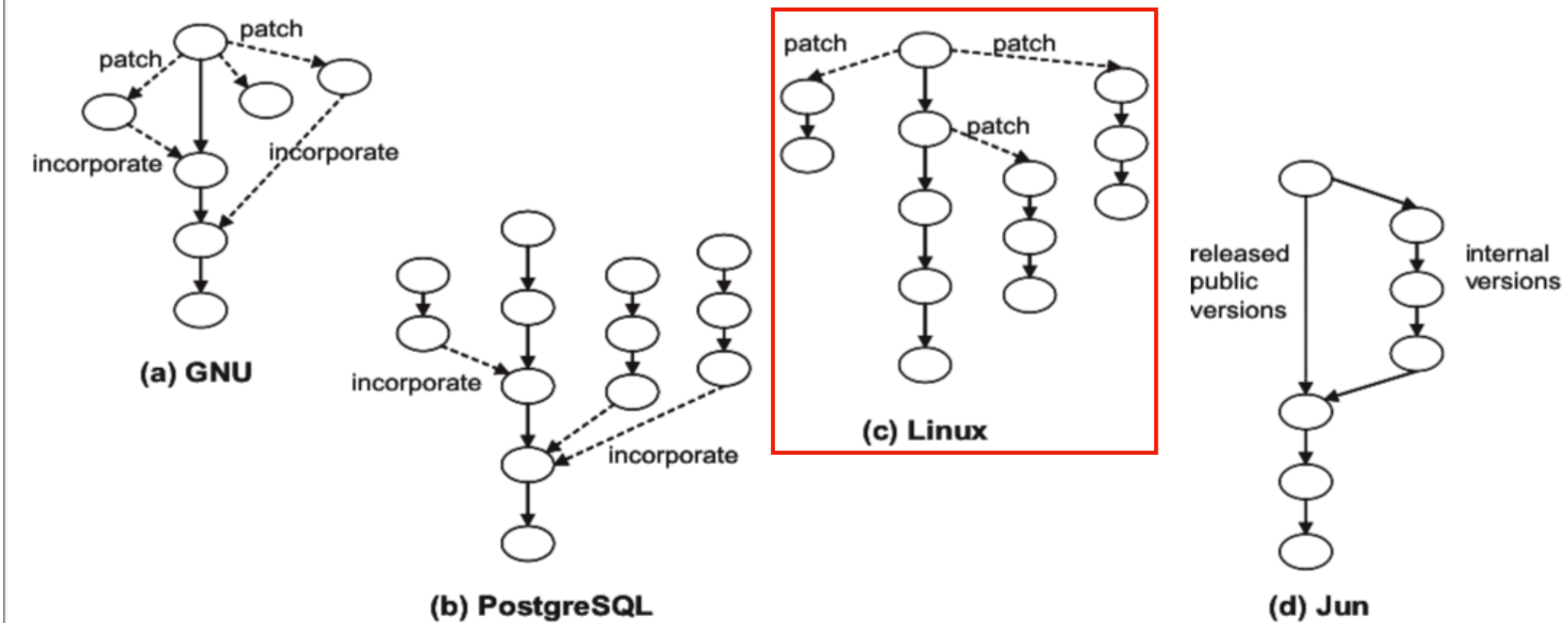
Figure 2: Evolutionary paths of the four projects.



2. Linux Support Project

- Linuxユーザのためにマスターシステムインテグレーターとしてサポートする
- 開発パッチへの返答の動機とモチベーションが低いため、同じ機能性のための複数の実装により単一のプログラムから多くのbranchが存在する

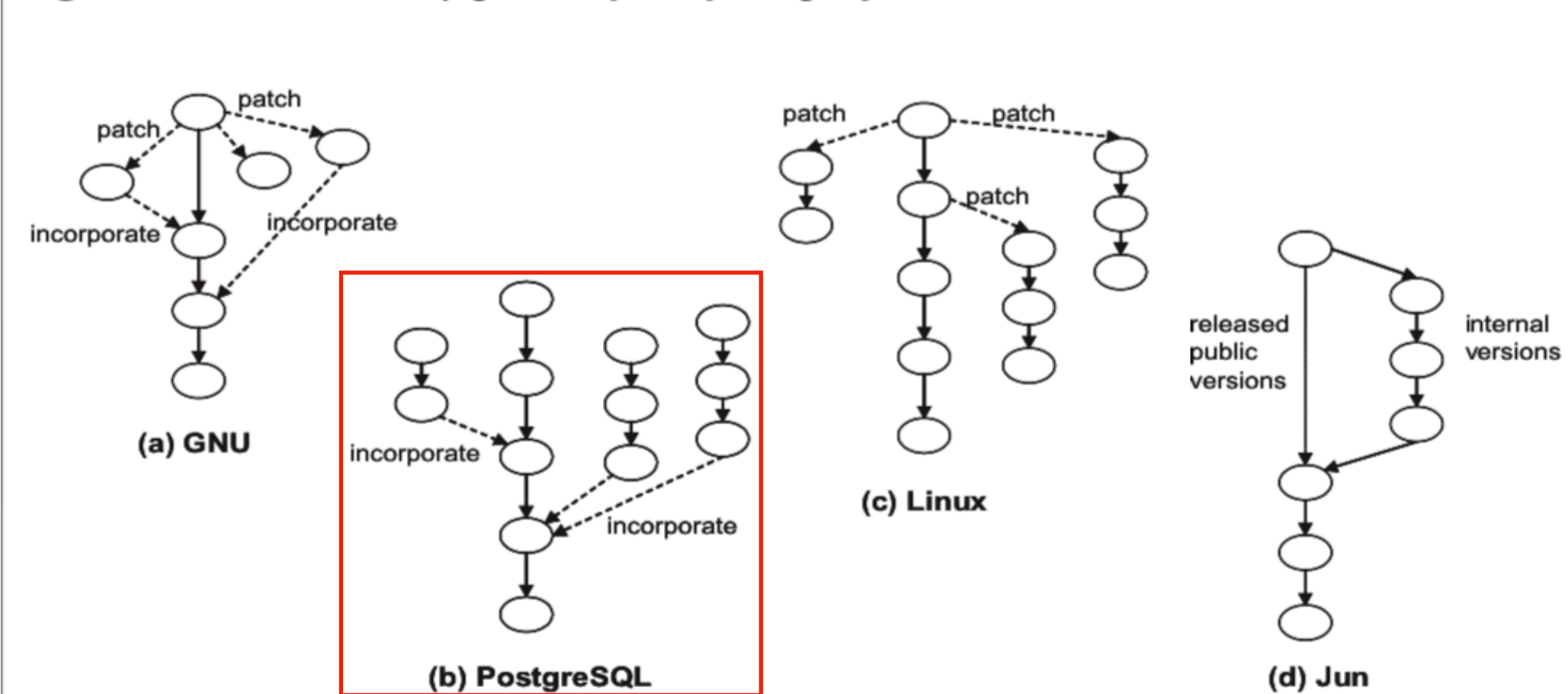
Figure 2: Evolutionary paths of the four projects.



3. SRA-PostgreSQL Project

- PostgreSQLデータベースを使う日本のユーザをサポートする
- 新規の要求に対してアクティブな開発者はそれらを満たすためのチームを編成するが、それらはパッチと比べて長く存在しコアメンバーによって承認された時のみコアバージョンに統合される

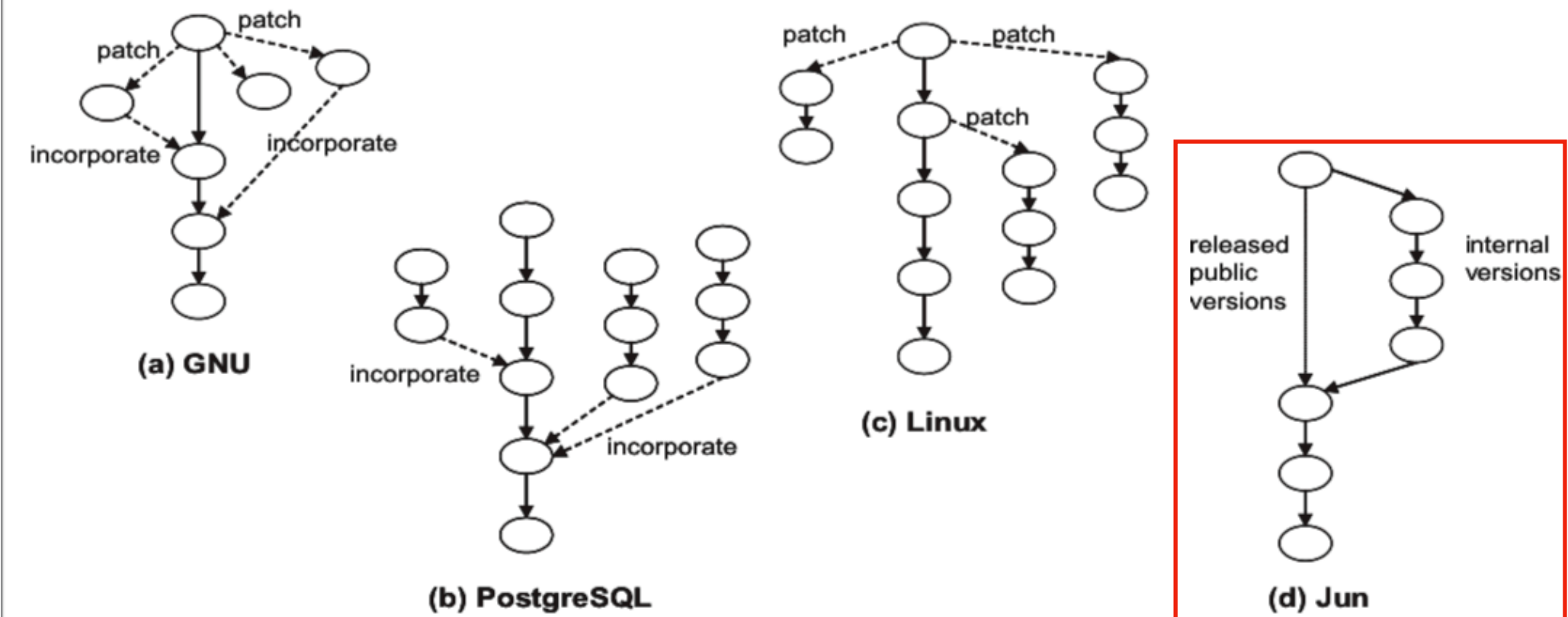
Figure 2: Evolutionary paths of the four projects.



4. Jun Project

- SmalltalkとJavaのための3Dグラフィックとマルチメディアのライブラリ
- 多くのOSSプロジェクト同様，内部仕様のために作成されたテストバージョンのブランチがたくさんあり，プロジェクトリーダーが十分テストされたと判断した時公式にリリースされる

Figure 2: Evolutionary paths of the four projects.



システムとコミュニティの進化

- システムの進化
 - バグ修正, バグ報告, パッチの統合など
- コミュニティの進化
 - 新たな貢献者の参加
 - コミュニティ内の貢献者の役割の変化

システムとコミュニティの共進化

- アクティブな貢献者は、その役割を変えることによってコミュニティを進化させるだけでなく、そのシステムを進化させるための源ともなる
- システムの進化は、バグ修正、バグ報告やパッチなどからコミュニティメンバーの役割を再定義することによって、コミュニティの進化につながっている



プロジェクトリーダーとコアメンバーはOSSシステムの進化だけでなく、新メンバーをシステムへ貢献させ、コミュニティの中心へと動かすような環境作りに焦点を当てるべきである

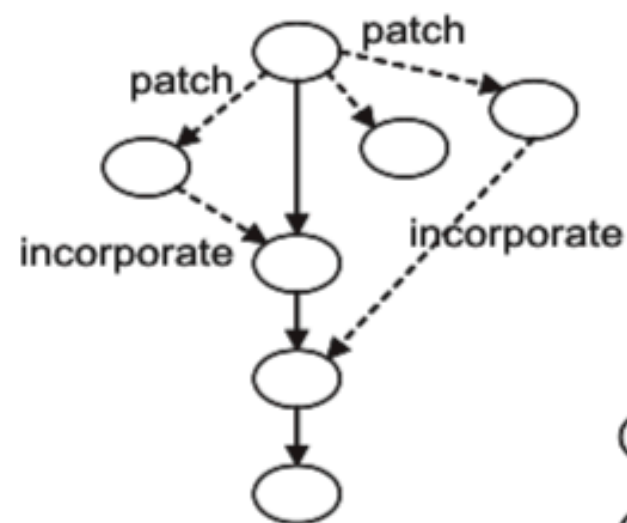
OSSプロジェクトの目的

- **Exploration-Oriented**（探索指向）
 - 新たなソフトウェアの開発に取り組み、**新手法と知識を共有する**
- **Utility-Oriented**（機能性指向）
 - 個人のニーズを満たすため、**機能性を充実させる**
- **Service-Oriented**（サービス指向）
 - OSSシステムの全てのステークホルダーに**安定したサービスを提供する**

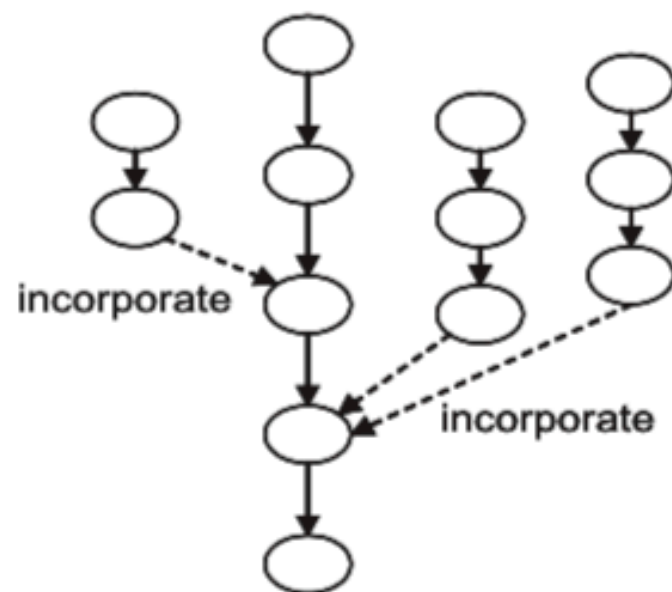
OSSプロジェクトの分類

Type	目的	管理体制	システムの 進化パターン	コミュニティ の構成	主な問題	例
Exploration -Oriented	新手法と知識 の共有	中央集権	コミュニティ からのフィー ドバックによ る単純な branch	Project Leader, Many Readers	プロジェクト の分離	GNU systems, Jun, Perl
	個人のニーズ を満たす	分散	トーナメント スタイル	Many Peripheral Developers	正しいプログ ラムの選択	Linux systems excluding the kernel
	安定したサー ビスを提供す る	中央会議	単純な branch	Core Members, Many Passive Users	保守性	PostgreSQL , Apache

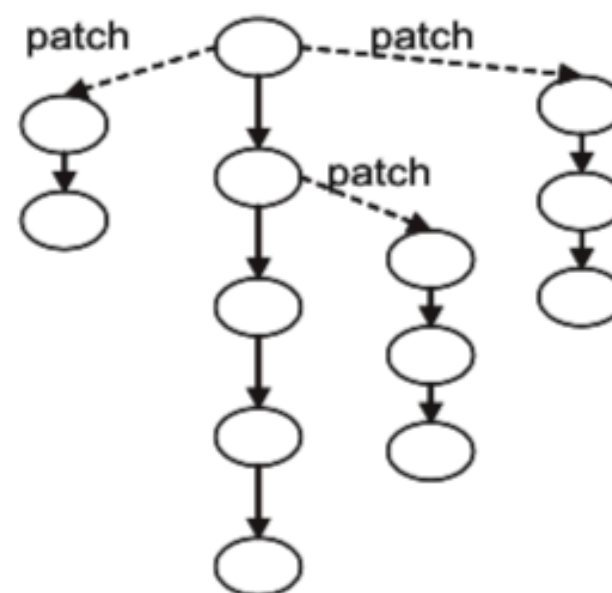
Figure 2: Evolutionary paths of the four projects.



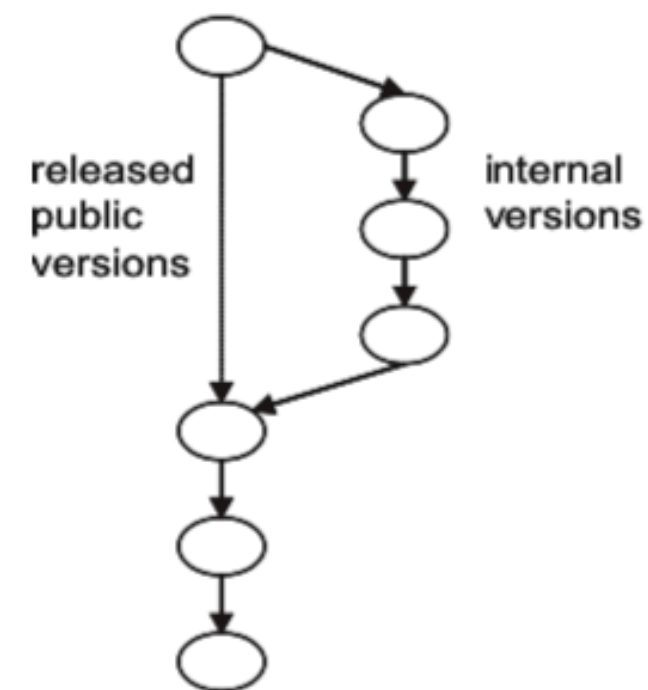
(a) GNU



(b) PostgreSQL



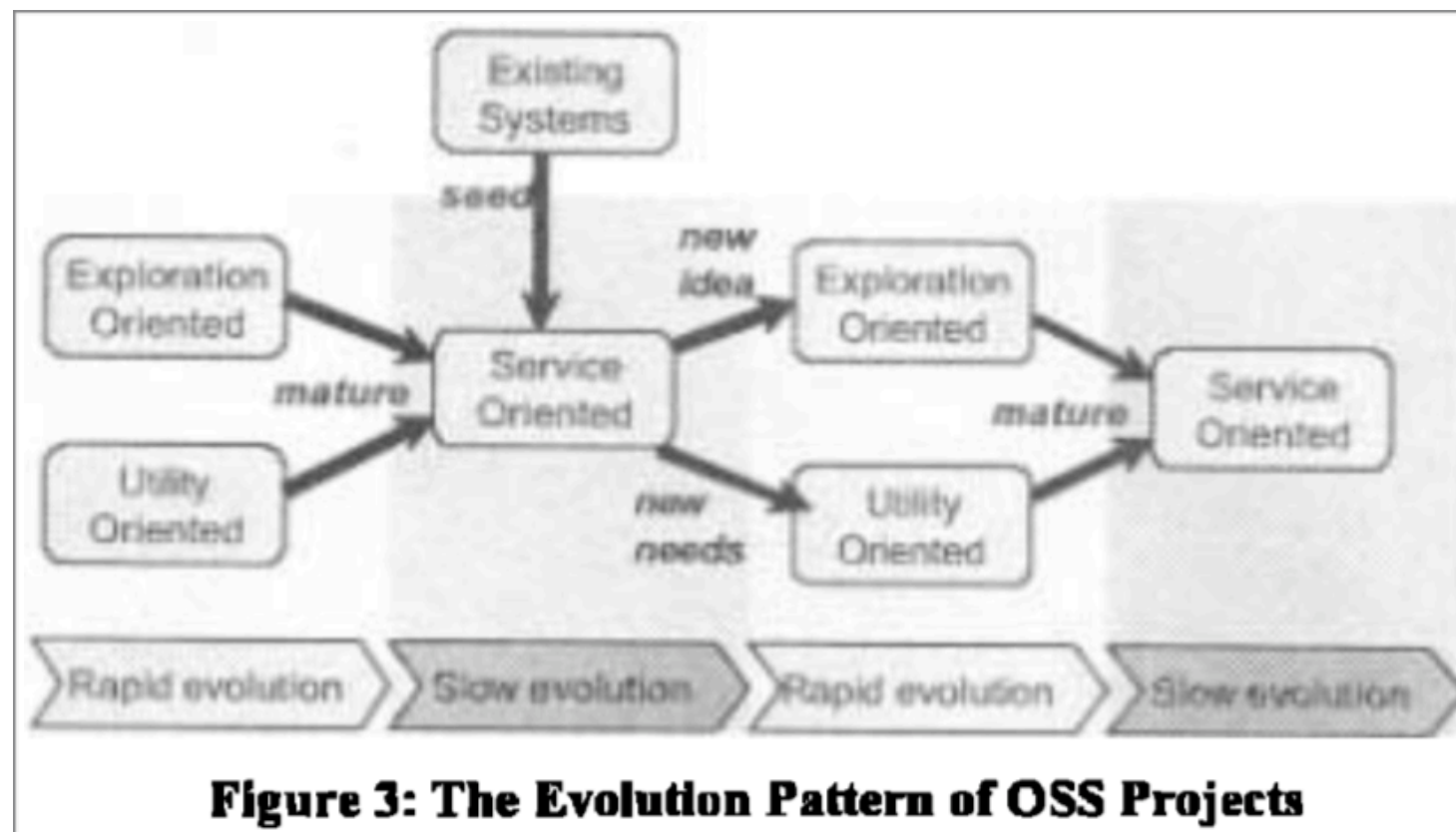
(c) Linux



(d) Jun

OSSプロジェクトの進化

- Exploration / Utility-OrientedはService-Orientedに進化する可能性がある
- 既存のシステムはさらなる成長のきっかけになり，ユーザがシステムを使うことで新たな要求やアイデアを想起させる
 - PostgreSQLは既存の調査実験に由来する



まとめ・所感

- まとめ

- OSSの長期間の成功にはシステムとコミュニティの共進化が必要である
- OSSプロジェクトにはそのプロジェクト毎に目的があり，システムやコミュニティの進化もそれによって異なる

- 所感

- OSS開発においてコミュニティの進化の重要性を知れた
- プロジェクトのタイプの違いによるシステムとコミュニティの共進化の違いが気になった

おしまい

