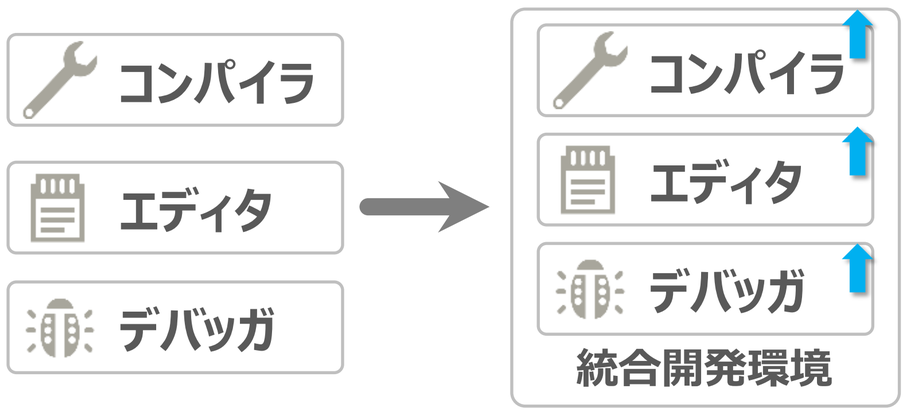
XDDPを加速させる
開発プロセスに則したソフトウェア構成管理

なぜソフトウェア構成管理なのか？

ソフトウェア構成管理は過去20年にわたり、**ブランチによるバージョン管理が主流**であり、根本的な考え方は変わっていません。ソフトウェア構成管理の運用に引きずられて、今のプロセスを変えることが難しい場合があります。

コンパイラなどの開発環境



コンパイラの性能向上，分散ビルドの技術革新など開発環境が進歩している

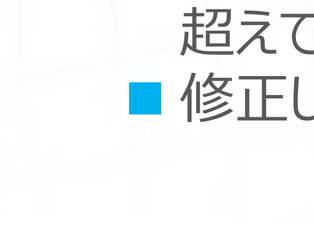
ソフトウェア構成管理

構成管理手法は進歩していない。未だにブランチ管理が主流であり、マージが開発者の業務として定着している

新たな取り組みを実施しようとしても、コードやドキュメントを管理しているシステムが古いままでは、取り組みを導入する際の障壁は取り除けません。

なぜ開発プロセスの変更が難しいか？

開発の現場において優先すべきは**開発ソフトウェアに直結する作業**であり、プロセス改善ではありません。
また、開発者は開発プロセスに関する技術スキルを習得しにくい傾向にあります。マネージャーや品質管理部門のように、品質改善に取り組む余裕がありません。



組織の運営、改善がミッション
- **プロジェクト全体の計画・遂行**
- **品質、要員、予算、進捗などの管理**

＝組織の改善が役割に含まれる

役割が異なる

開発の遂行がミッション
- **納期の厳守**
- **設計、実装、テスト等の実務スキルの向上**

＝組織の改善が役割に含まれない
改善のスキルは設計・実装・テスト等の実務スキルとは異なるため習得・着手しにくい

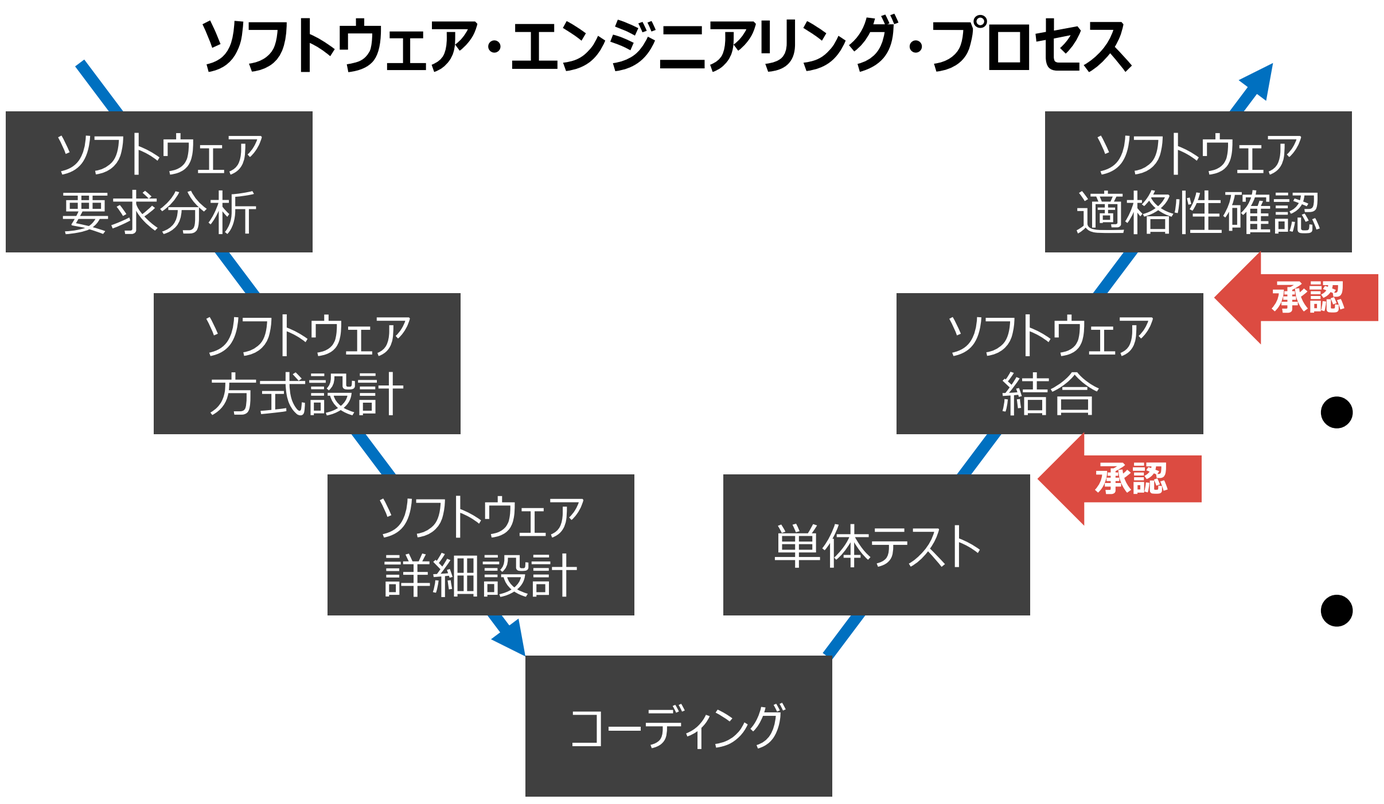
さらに、開発プロジェクト特有の制約から、プロセスを変更しにくい場合があります。
- 大規模な派生プロジェクトになるほど、各チームに開発プロセスを順守させることが難しく、またチームを超えて問題を共有するために多大な労力がかかる。
- 修正したバグをどの派生先へ展開すべきかわからず、チームリーダの経験にたよっている。

テクマトリックス株式会社 システムエンジニアリング事業部（ accurev-info@techmatrix.co.jp ）

TechMatrix

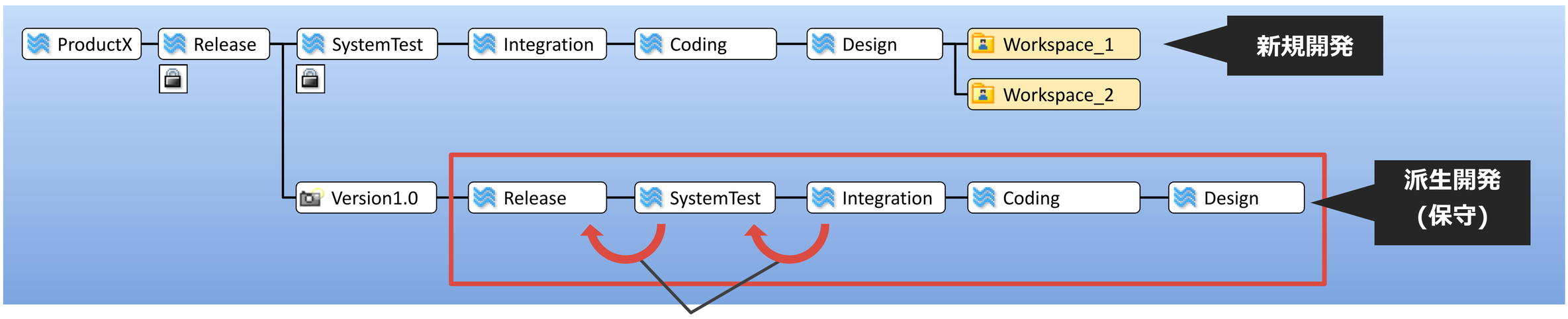
ソフトウェア構成管理の見直し＋プロセスの見える化

開発者が新たなプロセス・手法を容易に運用するために、ツール(AccuRev)による ①プロセスの見える化 を行うと同時に ②プロセスの変更に対応できる開発インフラを構築します。



- 開発プロセスは定義されているものの…
  構成管理とは別にプロセスを管理している。
- 開発者としてはついつい手元のコード中心に
  開発を進めてしまう。

ツールを使ってプロセスを見える化



Commit=承認として処理。プロセスの承認ルールと構成管理を一体化

アイコンの意味

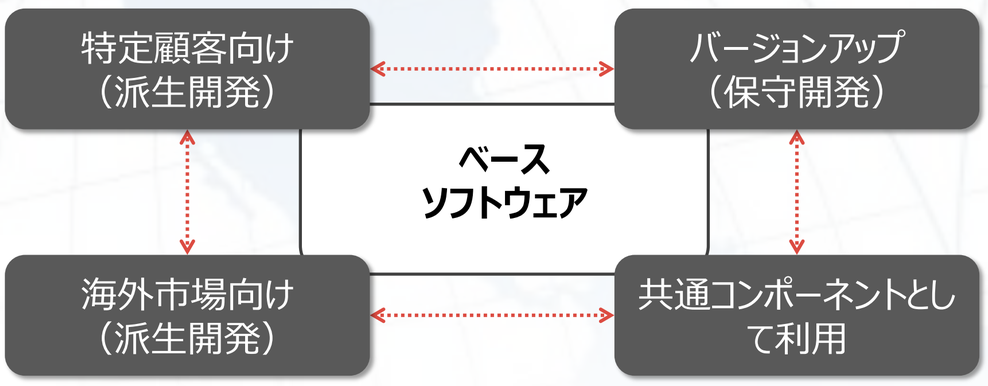
ストリーム（≒ 構成/ ブランチ）
- 製品の構成サブシステム、開発プロセス、開発チーム、 開発拠点などを表現

スナップショット （≒ タグ/ ラベル）
- リリースビルドなど任意の時点の状態を保管
- 過去の日時を指定することが可能

ワークスペース（ ≒ プライベートブランチ）
- 個々の開発者の作業領域

既存のソフトウェアへの変更の実施／他バージョンとのマージ作業

派生開発においては、従来のタグやラベル、ブランチといったソフトウェア構成管理では、頻繁なリリースサイクルに対応できず、リリースソフトウェア間の整合性を保つために多大な労力を要します。結果としてマージ漏れが発生し、品質の低下をまねく恐れがあります。

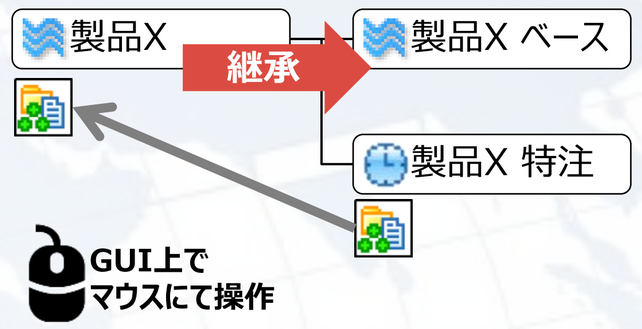


全てのソフトウェア間で整合性を保つことが求められます

継承

変更を上位のストリームにプロモートすると、下位のストリームに自動的に変更が反映されます。

関連する全てのソフトウェアに変更を反映させる場合に利用します。

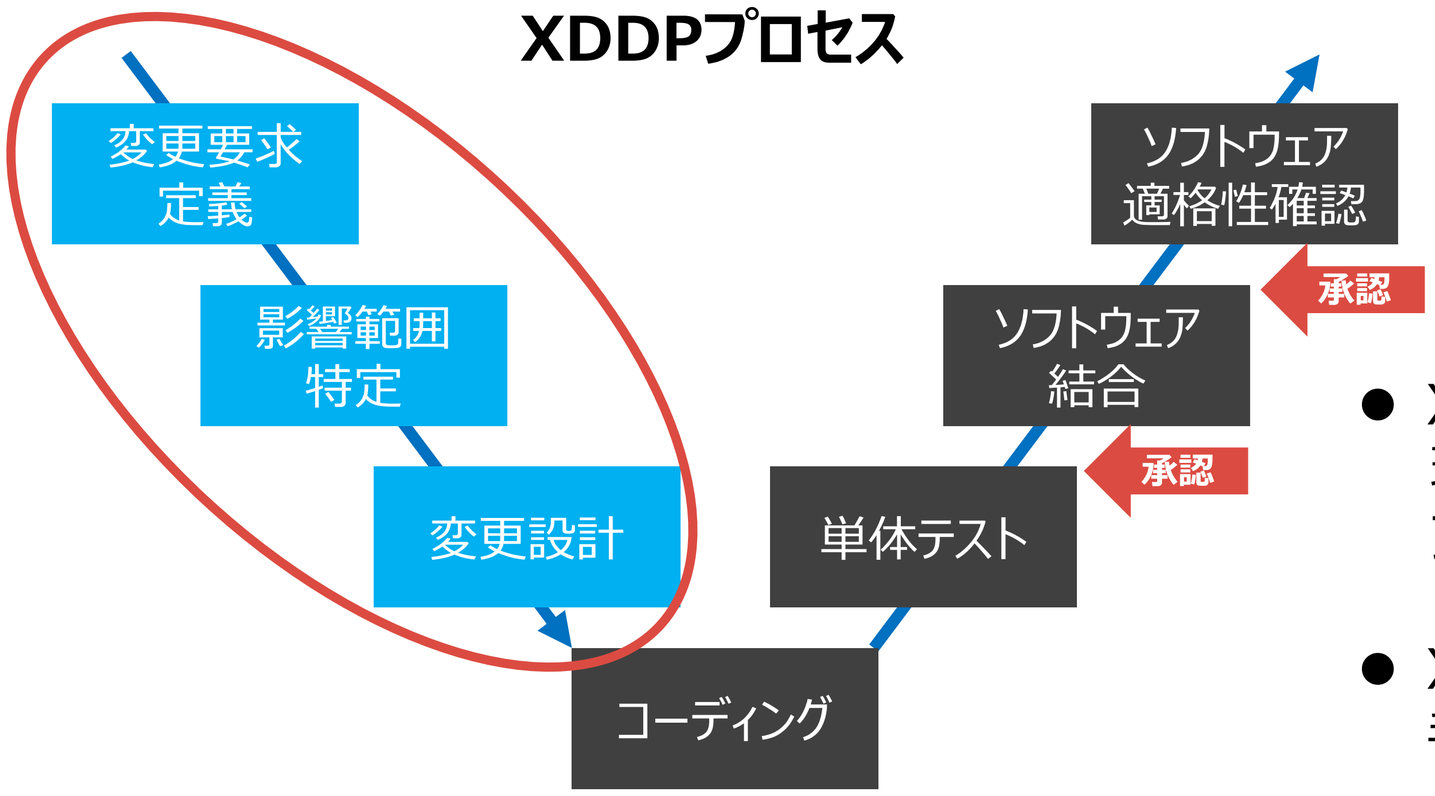


チェンジパレット

継承ではなく、特定の変更を特定のソフトウェアのみに反映したい場合に利用します。例えば、右図のように保守開発での不具合修正を次回の新規リリースにマージすることで、整合性を保ちます。

テクマトリックス株式会社 システムエンジニアリング事業部（ accurev-info@techmatrix.co.jp ）

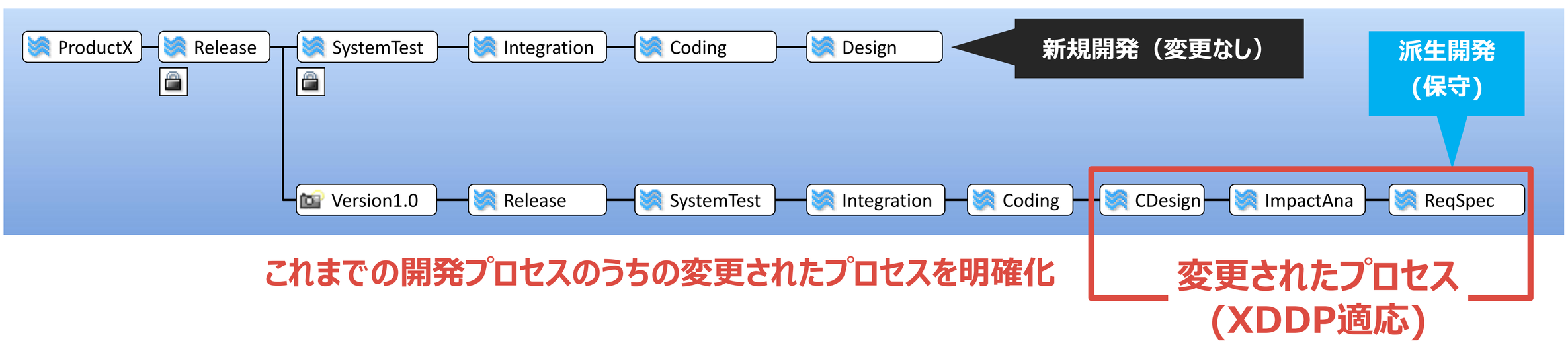
XDDPの適用



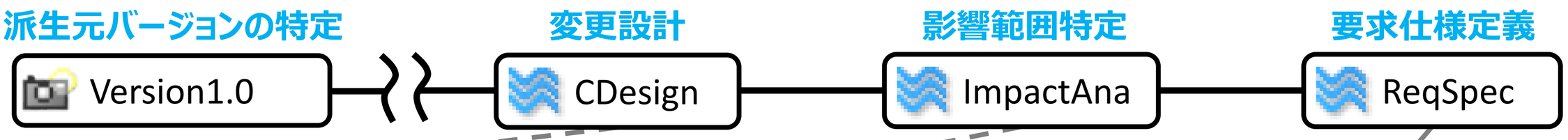
- XDDPを用いた開発を実施したいが…
  現在のプロセスを変更する際には、
  プロセスの管理を見直さなければならない。

- XDDPのプロセスを正しく運用できているかを
  手作業にて管理しているため、手間がかかる。

ツールを使ってプロセスを見える化



XDDPのプロセスをツール上に表現



成果物：変更設計書
変更前と変更後を記載します。他の成果物とともにパッケージ化されます。

成果物：トレーサビリティマトリクス
仕様とその変更箇所を特定します。
仕様やコードの変更はツール内でパッケージ化して管理されます。

成果物：変更要求仕様書
変更要求から仕様を定義し、各仕様はチケットとして登録します。

■ メリット

- プロセスの見える化：どの開発にXDDPを適用しているかすぐに把握できる。
- プロセスの順守　　：ソフトウェア構成管理にてプロセスを表現しているため、開発者は成果物を決められたプロセスへ登録するのみ。新しい取り組みを、人手の手間により達成しなくて良い。

- 開発プロセス・派生ソフトウェア・拠点等をツールにより管理し、まずは**開発者の負担を削減**します
- 開発者の余力ができた後、**段階的にXDDPを適用**していくことで派生開発を成功へと導きます