

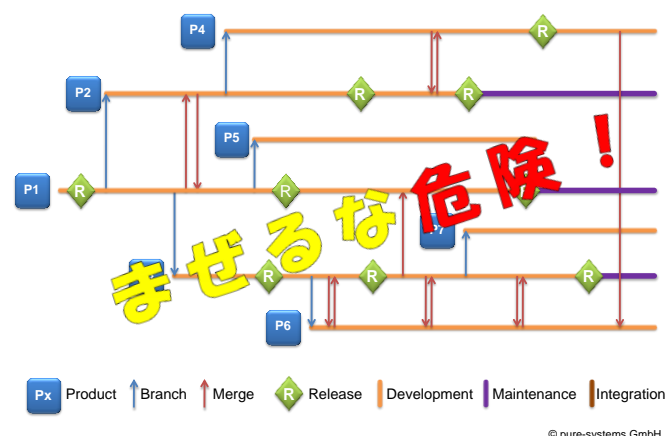
まぜるな危険！

プロダクトライン開発のバリエーション管理

プロダクトライン開発は、製品系列の資産の体系的な再利用に取り組むことで、派生製品の開発工数を削減して市場投入時期を早めることに加えて、品質改善や、製品間のトレーサビリティによるメンテナンス性向上などの相乗効果を上げる企業戦略である。

秘訣はバージョンとバリエーションの混同を避けること

プロダクトライン開発では、製品系列内で資産を共通要素と変動要素（製品間の違い）に分類して管理することで、体系的な再利用を目指す。その秘訣は、変動要素の管理（バリエーション管理）に、バージョン管理ツールを用いないことである。



バージョン地獄

上図は、典型的なバージョン管理から派生されるブランチ/マージのログで、複数の製品が既存システムのクローンから独自にメンテナンスされている。この状況では、全てのブランチされたインスタンスに変更を同期させ続けることは非常に厄介である。そして多くの場合2つの製品間のみでマージされることになり（いくつかのパーツを別の製品から取ってくるだけ）、体系的な再利用にはならない。バ

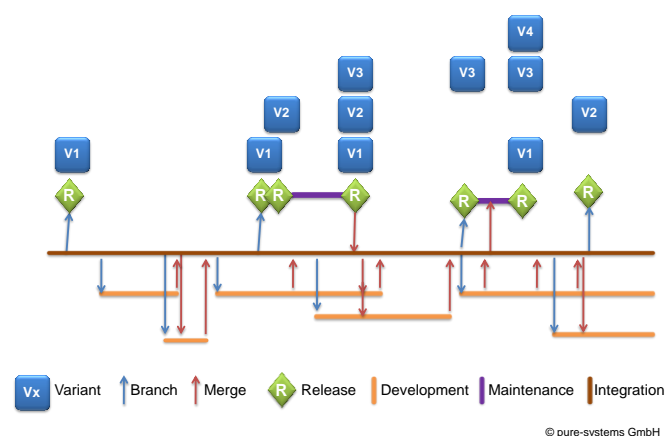
ージョン管理される資産の粒度が変動要素と同等で無い限り、バージョン管理内のブランチは変動要素の表現には向かない。そしてファイルの資産に関しては、粒度のミスマッチは全く避けることができない。資産への変更をそのライフサイクルにわたって追跡するために適正なバージョン管理ツールは必要であるが、バリエーションの管理を一緒にすることはできない。

再利用の課題

- ・機能追加で変動要素、依存関係、複雑性が指数関数的に増える
- ・既存ツールではバリエーション管理をサポートしない
- ・製品開発ライフサイクルを通して資産の体系的な再利用ができない

バリエーション管理

下図は、共有資産から製品バリエーションが生産される様子を示す。バリエーションの管理は、pure::variantsのような適正なツールによって、独立した活動として実施されるべきである。バージョン管理システムはインスタンスの記録に使われるが、バリエーション管理の仕組みは提供しない。



https://www.fuji-setsu.co.jp/products/purevariants/Danilo_Blog.html