

名称	概要	対応できる粒度	留意点	利点	バリエーション範囲	Direct Link to Use of Assets	必要なツール	Explicit Process / Evolution Support
パラメータ付け	可変パラメータは制御フローとアセットの挙動を変更するために使われる。	細	パラメータの依存性が数学的に証明可能な形式で明確に表現されることは通常はあり得ない。パラメータ付けされたコードは特定のバリエーションに適合させながら読まなければならないので複雑になりがちである。	効率 (#ifdef) 動的変更 (if (param == 12))	構造(すべての粒度) パラメーター	×	パラメーター付けの概念を持つツール(制御構造、プリプロセッサ、…)	×
コンポーネント	念入りに定義されたインターフェースを持つコンポーネントがシステムに組み込まれる。	粗(インターフェースまたはコンポーネントあるいはその両方のレベル)	既存のコンポーネントを修正する際は、すでにそのコンポーネントを使っている部分に影響を与えないようにする。しかし多様な使用方法に対応させようとして、しばしばインターフェースや関連コードを肥大させてしまう。あるいはコンポーネントを複雑にし、保守を困難にし、メモリ使用量を増やしてしまう。インターフェースにバージョンを付けることで保守が必要なアセットを増やし、構成手順を複雑化させてしまう。バリエーションを扱うには内部パラメータ付けと組み合わせることが多い。	念入りに定義されたインターフェースによって開発工程と分離されるのでテストが簡素化される。安定したソフトウェア部品によって再利用は容易になる。	構造(コンポーネント)	×	特別なツールなしでファイルシステムによる管理だけでも構築できるが、AUTOSAR準拠ツールのように特化したアプローチもあり得る。	×
クローンアンドオウン ブランチ	リポジトリから適切なアセットが選択され、コピーやブランチされて要求仕様の変更に沿って修正される。	粗(ブランチやコピー可能なアセット、通常はファイル単位以上)	多くの場合、アセットのコピーとブランチはバリエーションの最小粒度に適合しないので、コードやその他の情報が重複して保守の手間を増やしてしまう。またそれは最利活用可能と思われるアセットのサイズを増やし、選択と構成を徐々に複雑化していく。あるいは独立して個別に進化したアセットの機能のマージは入り組んでいるために再利用を複雑化させる。	開発ラインの分離によって、他に影響を与えず迅速な変更が可能になる。	構造(アセット)	○	ファイルシステムによる管理、バージョン管理システム。	△
バージョン付け	変更されたアセットはバージョン番号によって識別される。システムは必要に応じてアセットとバージョン番号の有効な組み合わせを選択することで構成される。	粗(バージョン管理可能なアセット、通常はファイル単位以上)	バージョン番号は単なる名前に過ぎず、バリエーションの内容に関する情報を含まないため、適切なバージョンを選択するためには別の追加情報が必要になる。各バージョンのアセットは原理的に再利用可能なので、どんどん数が増えていく。その結果コピー方式に頼ることになり、再利用性の悪化を招いて保守の手間を増やす。	開発ラインの分離によって、他に影響を与えず迅速な変更が可能になる。	構造(バージョン付け可能なアセット)	○	手法次第。通常はバージョン管理システムなど。	△
フレームワーク	フレームワークはアプリケーションによって使われる定義されたサービス群を提供する。	可変	フレームワークは全体としてテストされリリースされる。フレームワークのバリエーションが最小に保たれるならば、高い再利用性が得られる。	柔軟性を得ようとする、複雑な設定やフレームワークのパラメーター付けが必要になることが多い。フレームワークを常に使い切ろうとすると、パフォーマンスとメモリーサイズの点でたい他の手法に劣る。通常は設計から実装の段階で使われ、要求分析には用いない。	パラメーター 構造(粒度に依存。通常はフレームワーク)	×	フレームワーク次第	×
ドメイン特化言語	低級汎用言語を抽象化した「状態」や「状態遷移」のようなアプリケーション固有の構成を可能にするドメイン特化言語	可変(ドメイン特化言語の仕様次第で変わる)	念入りに定義されたドメイン特化言語は、管理するアセットのサイズを小さくしてその理解を簡単にする。また変化点の概念を持ち、再利用の複雑さの増大に対処する方法をユーザーに提供することもできる。	業務にマッチしたドメイン特化言語の設計及び保守のノウハウと投資を必要とする。通常は設計から実装の段階で使われ、要求分析には用いない。	すべて(ドメイン特化言語に依存)	×	ドメイン特化言語の開発ツールとその言語仕様から生成されたツール	×
プロダクトライン開発	再利用可能なアセットから派生したバリエーションの明確な概念を与え、開発工程全体を通して統一されたバリエーションの表記を使って再利用の概念を結合することで、種々の再利用の手法を包括的な視点に統合する。アセットの進化を扱うプロセスを含む。	可変	管理するアセットの数とサイズが大幅に削減されるため、プロセスを簡素化し、保守の手間を省いて再利用性を向上させる。バージョン管理も簡単になる。	コラボレーションはより複雑になる。再利用におけるアセットの使用の高度な結合。品質保証の向上の可能性。	すべて(何でも)	○	必須ではないが、バリエーション情報の保守とバリエーションの統合のための専用ツールがよく使われる。	○