

商品	XDDP	確認質問	ソフトの派生開発を行っていますか？
ターゲット	派生開発において納期遵守に困っているソフト開発関係者		品質低下や納期遅延で困っていますか？

関係者	I : 問題について合意する	KEYWORD	III : 解決方法が、問題を解消させることについて合意する	KEYWORD	IV : 重大なマイナスの影響を生まないことについて合意する
問題質問	開発者、マネージャー 後工程やリリース後にデグレードや変更間違いによる手戻りが多いですか？ 【補足】手戻りの具体的な内容(変更要求の間違い, 変更箇所の間違い, 変更方法の間違い)を質問するとよい。変更3点セットは手戻りの内容それぞれに対応する(変更要求=USDM, 変更箇所=TM, 変更方法=変更設計書)。	デグレードと変更間違いによる手戻り	レビューが十分にでき、コーディング後の手戻りが少ないと良いですか？ 【補足】XDDPは、レビューを中心とした手法であり、それを実現するため、コーディングを留保する。留保で得られた時間をレビューに当てることで後工程での手戻りが減ることになる。	コーディングの留保	工数が増える/追加の作業が増える ・レビューによって手戻りが減るので、工数増加はしないというロジック/事例を説明する ・PFDによりプロセス・ドキュメント体系をプロジェクト毎に適切に設計するので、作業の無駄が減る ・スモールスタートで検証してみる
	開発者、品質保証部 開発プロセスが実情にあっておらず、無駄だと感じることがありますか？ 【補足】組織標準などがある場合、無駄なドキュメントが作られている可能性がある。開発途中で新たにドキュメントが必要になったりした場合も同様(冗長になりがち)。	開発プロセスの無駄	派生開発に適したプロセスで無理・無駄がないと良いですか？ 【補足】XDDPは、レビューを中心とした派生開発に特化した形のプロセスを提供する。また個別にプロセスを設計しなおすため、無理や無駄が減る。	変更用プロセス	コーディングを留保しすぎて、納期を守れなくなる ・サイズ見積りに基づく、工程見積りにより、コーディング開始時期を明確に定義する。これにより納期を守れないという状況は発生しづらいことを説明する
	開発者、マネージャー (時間がないや納期が恐いなどの理由で)ソースコード変更による影響分析が担当者任せになっていませんか？ 【補足】一人プロジェクト(担当者丸投げ)になっていないかどうか。十分に精査せずに変更箇所を見付け次第修正するやり方になっているかどうか。	担当者任せの変更精査	変更方法が十分かつ効果的に設計・レビューされていると良いですか？ 【補足】XDDPでは、効果的なレビューのためのドキュメントを準備してある。そのドキュメントは「変更」における必要最小限(変更要求(What/Why)=USDM, 変更箇所(Where)=TM, 変更方法(How)=変更設計書)のものである。	変更用ドキュメント(変更3点セット)	自組織の開発に適さない可能性がある/本当に自組織で効果があるかわからない ・スモールスタートで検証してみる ・既存開発のデータを用いて、効果をシミュレーションしてみる
					失敗のリスクがある ・スモールスタートで検証してみる ・既存開発のデータを用いて、シミュレーションしてみる
重大質問	開発者、マネージャー、経営者、顧客 「楽勝!」と思われていたささいな変更でも納期に間に合わないことが多いのではないですか？ 【補足】「開発遅延による強引なリリース」でも可。「開発後半で火を吹く」という状況かどうか	納期遅延	見積り通りに開発が終了すると良いですか？ 【補足】XDDPは、変更3点セットの各所でサイズ見積りの見直しを実施する。それを基にして工数を見積り、コーディング開始時期を決定する。コーディングまでの時間を品質担保に使っているため大きな混乱なくプロジェクトが終了する。	見積り通りの開発	過大な効果を期待してしまう ・過大な期待をさせないように、トップ、マネージャーに正確な情報を入力する ・スモールスタートで早期に適用効果を見積る
	開発者、マネージャー (既に変更してしまった場所についてより良い実現方法を後から思いつくなどした場合に)本当は綺麗なコードを書きたいのに時間がなく、小手先の変更をせざるを得なくなり、開発者のモチベーションが低下してませんか？ 【補足】自己欺瞞(自分で自分の心を欺くこと)をせざるを得ないかどうか。自己欺瞞は、心理的に不健全な状況を生む。	自己欺瞞	開発者が開発の意義を感じていると良いですか？ 【補足】コーディング開始までに可能な限り最良と思われる実現方法を模索するため、小手先の対応(自己欺瞞)になることはない	モチベーション向上	定着しない ・エバンジェリストを育成する ・組織的に定着化を図る ・トップダウンで適用を宣言する
	開発者、マネージャー、経営者、顧客、品質保証部 小手先の変更が積み重なって、ソフトの品質がどんどん劣化していませんか？ 【補足】上記の状況が続くことで、ソースコードが汚くなっていないかどうか	ソフト品質の低下	ソフト品質が維持/改善していると良いですか？ 【補足】「改善」は意図的にリファクタリングしなければならないが、「劣化」を防ぐことはできる	ソフト品質の維持/改善	成功後に他プロジェクトに巻き込まれる ・トップ、マネージャーに他プロジェクトに巻き込まないという確約をもらう
	経営者、顧客 納期遅延(品質低下、コスト超過など)が原因で顧客の信頼を失っているのではないですか？また、そのことで会社業績を圧迫していませんか？ 【補足】実際の経営者が気にしそうな内容を記載。	顧客信頼の失墜	納期通り(高品質など)により、顧客の信頼を得ることで会社の業績が上がっていると良いですか？ 【補足】重大質問と対応させる	顧客信頼の回復	

II 問題解決の方向性について合意する【ポジショニングトーク】	提案する商品と想定顧客	コンセプト説明文
XDDPとは、派生開発において、品質が低下し、納期も守れなくなるという問題に対処する手法。従来の変更箇所を見付け次第変更するといった開発とは異なり、コーディングを留保し、その間で徹底的にレビューを行うことで、手戻りがなくなるため、納期も守りながら品質も維持することが可能になる。そのための効率的なドキュメント(変更3点セット)や変更プロセスを含んでいる。	XDDP 派生開発において納期遵守に困っているソフト開発関係者	デグレードや変更間違いによる手戻り、開発プロセスの無駄、担当任せのソースコード変更精査といった問題があるとのことで、XDDPは、コーディングの留保、変更専用のプロセス、変更専用のドキュメント(変更3点セット)により、解決することが可能です。これらの問題が解決できれば、ソフト品質低下、納期遅延、モチベーションの低下といった重大な問題を解決できます。さらにそのことで、来たるべき新規開発への備えも可能になります。

対立	V : 導入を阻害する障害を克服する方法について合意する
	社内関係者の合意を得る ・社内関係者(主にユーザ、決済者)に対して、本マフィアオファースシートを用いて合意を取る ・対象者に合せてマフィアオファースシートをカスタマイズする
	社外関係者の合意を得る ・社外関係者(主に関係者)に対して、本マフィアオファースシートを用いて合意を取る ・Win-Winになるような方法の検討のために、対象者に合せてマフィアオファースシートをカスタマイズする
	組織標準や従来のやり方との対応をとる ・組織標準のドキュメントやプロセスとの対応関係を取る(USDMは〇〇仕様書に対応する、など) ・XDDPを組織にテラリングした事例を参考にする
	導入工数を確保する ・工数/予算の決定権のある人物にXDDPをプレゼンして、工数/予算を貰う ・スモールスタートで検証して、必要工数/コストを見積る ・既存開発のデータを用いて、擬似的に検証し、必要工数/コストを見積る
	スキルを習得する ・AFFORDD主催の勉強会に参加する ・独自の勉強会を開催する ・エバンジェリストを置いて、展開を推進する
	パイロットプロジェクトで試行する ・やる気のある開発者がいて、社内で説得力のあるプロジェクトをパイロットプロジェクトとして、XDDPを試行してみる ・試行が不可能である場合、ながら作業表を用いて、過去プロジェクトのデータからXDDP導入時の効果を見積ってみる