



初めてのプロジェクト 発注ガイド

成功するプロジェクトのために

このガイドの内容

- | | |
|--|---|
| <ul style="list-style-type: none">1. 初めてのプロジェクト発注ガイドについて2. 発注前に準備しておきたいポイント3. プロジェクト発注の基本フロー4. アジャイルとウォーターフォールの違い5. 開発の流れ6. 成功するプロジェクトのポイント7. トレードオフスライダーについて8. 格安で開発することのデメリット9. システム開発以外の予算も考慮する10. ヒアリングでよく聞かれる質問例11. テストの重要性12. トラブルを防ぐために13. よくある質問(FAQ)14. GRILL のサポート内容15. お問い合わせはこちらから | <ul style="list-style-type: none">ー ガイドの目的と概要ー プロジェクト開始前に整理すべき事項ー 発注から納品までの流れー 開発手法の特徴と選び方ー 要件定義から納品までの各フェーズー 発注から実現までの重要な要素ー プロジェクト優先事項の明確化ー 安価な選択のリスクを理解するー 事業成功に必要な費用の全体像ー 初回ヒアリングで準備すべき内容ー テストがプロジェクトに与える影響ー トラブルを回避する具体的な対策ー よく寄せられる疑問とその回答ー 当社が提供する支援と特徴 |
|--|---|

1. 初めてのプロジェクト発注ガイドについて

「初めてのプロジェクト発注ガイド」は、プロジェクト発注の経験が少ない方や初めて外部への依頼を検討している方を対象に、スムーズに発注プロセスを進めるための基本的な流れやポイントをわかりやすく解説することを目的としています。

また、このガイドを通じて、発注者が適切な準備を行い、**プロジェクトを成功へ導くための知識と具体的なヒント**を提供しています。

本ガイドの内容は、株式会社GRILLに帰属します。社内資料としての使用：貴社内での参考資料としてご利用いただけます。再配布・転載の禁止：本ガイドの内容を無断でコピー、再配布、または第三者に提供することは禁止です。

ご利用に際して不明点や特別な許可が必要な場合は、株式会社GRILL までお問い合わせください。

2. 発注前に準備しておきたいポイント

事前にプロジェクトの内容を、チェックリストで整理しましょう。

□ プロジェクトの目的と背景を整理する

- 解決したい課題は何ですか？
- 最終的に目指したいゴールは何ですか？

□ 想定するユーザーを考える

- 誰がこのサービスを使いますか？
- ユーザーが期待する便利な機能は何ですか？

□ 必要な機能と優先順位を決める

- 必須機能(例:決済機能、データ分析機能)
- あれば良い機能(例:SNS連携、デザインカスタマイズ)
- 将来的に追加したい機能(例:多言語対応)

□ スケジュールと予算を検討する

- 納期までの期間にどのくらい余裕がありますか？
- 初期費用だけでなく、運用コストも検討しておきましょう。

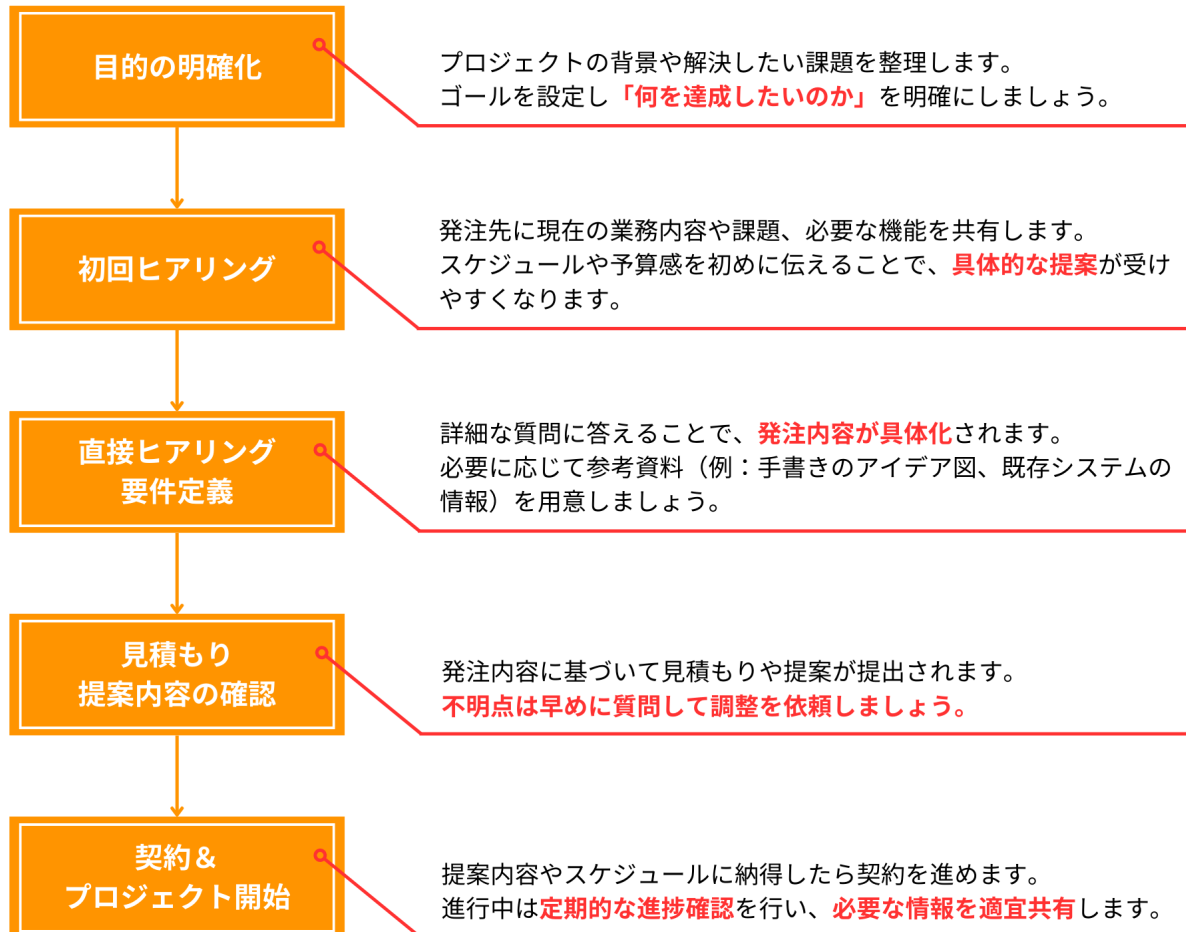
プロジェクト発注の成功は、事前準備の精度に大きく左右されます。「発注前に準備しておきたいポイント」を確認することで、以下のようなメリットが得られます。

1. 課題や目標が明確になる
2. 効率的なヒアリングが可能となる
3. コストとスケジュール管理が容易になる
4. リスクの低減につながる
5. プロジェクト全体の見通しが立てやすい

代行業者を探す前、依頼を行う前の段階にこのような事前準備を徹底することで、プロジェクト全体の見通しが立ち、成功への土台を築けます。

3. プロジェクト発注の基本フロー

プロジェクト発注までの基本フローを理解し、成功までの道筋を理解しておきましょう。



- それぞれのプロセスにかかる期間は、プロジェクトのボリュームによって異なります。
- 必ず直接ヒアリングの場面で、およそのスケジュール期間を確認する様にしましょう。

4. アジャイルとウォーターフォールの違い

アジャイルとウォーターフォールは、開発プロジェクトの進め方として異なる特徴を持っています。

まず、ウォーターフォール型は、各工程を順に進める計画的な手法で、要件が明確で変更が少ない場合に適しています。

ただし、途中での変更に対応しづらい点が課題となります。

一方、アジャイル型は、小さな単位で開発を進めながら改善を重ねる手法で、変化に柔軟に対応でき、ユーザーのフィードバックを反映しやすい特徴がありますが、全体像が見えにくいことがあります。

これらの特性を組み合わせたハイブリッド型もあり、プロジェクトに応じて最適な手法を選ぶことが重要です。



ウォーターフォール型

開発工程を順番に進める手法

設計から実装
テストまでを段階的に進め
次の工程に進む前に
前の工程を完了する



アジャイル型

小さな単位で 開発を繰り返す手法

数週間ごとに
小さな成果物を作り
改善を繰り返して 開発を進める



ハイブリッド型

両者を組み合わせた手法

設計や要件定義は
ウォーターフォール型で
実装や改善はアジャイル型で進める

手法	メリット	デメリット
ウォーターフォール	設計が明確で 進捗管理がしやすい	仕様変更に対応しにくい
アジャイル	柔軟に変更に対応可能 ユーザーのフィードバック を活用	全体像が見えづらく 進行管理が難しい場合がある
ハイブリッド	各工程の 良い部分を活用できる	管理が複雑になる場合がある

5. 開発の流れ

プロジェクト開発は以下の流れで進行します。要件定義から納品までの各フェーズについて知っておきましょう。



6. 成功するプロジェクトのポイント

成功するプロジェクトには特徴があります。しっかりポイントを押さえておきましょう。

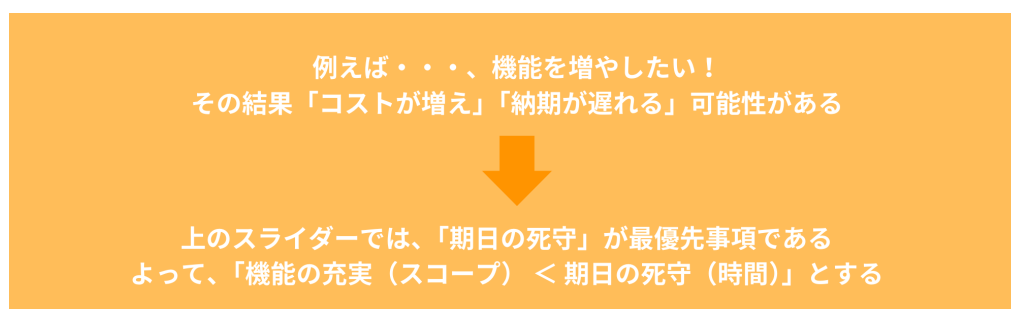
POINT 1 INFORMATION	POINT 2 COMMUNICATION	POINT 3 ADJUSTMENT
情報共有を丁寧に行う	適切な コミュニケーション	柔軟な対応を心掛ける
発注先にはできるだけ詳細な 情報を伝えましょう。 図解やサンプルも役立ちます。	定期的な進捗報告や質問への 迅速な対応が、スムーズな進 行につながります。	必要に応じてスケジュールや 要件を調整しながら進めるこ とが重要です。

7. トレードオフスライダーについて

プロジェクトで優先する事項を明確化しましょう。以下がトレードオフスライダーの一例です。

スコープ	項目
MIN ← ● → MAX	スコープ：機能を全部揃える
MIN ← ● → MAX	予算：予算内に納める
MIN ← ● → MAX	時間：期日を死守する
MIN ← ● → MAX	品質：高い品質、少ない欠陥

プロジェクトには上記の例のように「機能の充実」「コスト削減」「納期短縮」のようなトレードオフが存在します。



このようにトレードオフスライダーを活用し、プロジェクトで何を優先すべきかを発注者と開発者が共通認識を持つことが重要です。

8. 格安で開発することのデメリット

プロジェクトの予算を過度に抑えることによって生じる、デメリットについて確認していきましょう。

1. 予算不足による追加請求

- 初期費用が安く見えても、途中で予算が足りなくなり追加請求される場合がある

2. 品質が低い

- 設計やテストに十分な時間がないため、運用後のトラブルになるリスクがある

3. 納期が遅れる

- 開発の進行が遅れることで、予定していたサービス開始に影響が出ることがある

4. 対応力の不足

- 不具合発生時や急な変更に対して、柔軟な対応が期待できない場合がある

相見積もりを行う際、価格だけでなく、対応力や品質、サポート体制も確認することが非常に重要です。

9. システム開発以外の予算も考慮する

プロジェクトは、「開発終了」で完了になる訳ではありません。



単なるシステム開発ではなく、事業成功を目指したパートナーシップを構築できる開発業者を選ぶことが重要です。

10. ヒアリングでよく聞かれる質問例

質問と回答を事前に想定しておくことで、スムーズな交渉が可能になります。

よくある質問1

どのような課題を解決したいですか？

回答例) 作業が煩雑である、顧客対応に時間がかかるなど

よくある質問2

必要な機能を具体的に教えてください。

回答例) データ管理、ユーザー認証、モバイルアプリ対応 など

よくある質問3

対象となるユーザー層や利用シーンを教えてください。

回答例) 学生向け、営業担当向け、スマホ利用が主 など

よくある質問4

必要な機能を具体的に教えてください。

回答例) ○○のデザインが好き、○○のツールの機能を発展させたい

よくある質問5

スケジュールや予算の希望はありますか？

回答例) 「来年○月までに完成」「初期費用は○○万円以内で」など

11. テストの重要性

テストは顧客側でも重要な作業

- ユーザー視点で実際にシステムを動かし、不具合や期待と異なる動作がないか確認します。
- 顧客側のテストが遅れると、納期にも影響が出ます。

仕様変更や機能追加の影響

- 仕様変更や機能追加を行うたびに再テストが必要です。
- 無計画な変更や追加は、開発コストやスケジュールに大きな影響を及ぼします。

慎重なテスト計画

- バグの早期発見と修正により、運用開始後のトラブルを防ぎます。

テストは、システムやアプリの品質を確保し、期待通りの動作を保証するために欠かせない工程です。開発者だけでなく、発注者である顧客側も積極的に関与することが求められます。特に、最終的なユーザー視点での検証は、システムが実際の使用環境で適切に動作するかを確認する上で重要です。

また、仕様変更や機能追加が行われるたびにテストが必要になります。この際、テストがスムーズに進まないと、プロジェクト全体の進行が遅れ、納期にも影響が出る可能性があります。そのため、変更や追加のタイミングには慎重な検討が必要です。

さらに、テスト計画の不備や省略は、運用開始後の重大なトラブルや修正コストの増大を招くリスクがあります。プロジェクトの成功には、テスト工程を単なる確認作業としてではなく、システムの品質保証プロセスとして位置付け、十分な時間とリソースを確保することが不可欠です。

テストを徹底することで、運用開始後のトラブルを未然に防ぎ、システムの信頼性を高めることができます。発注者と開発者が連携し、計画的かつ効果的にテストを進めることが、プロジェクト成功の鍵となります。

12.トラブルを防ぐために

⇒ 要件や契約内容は明確に記録する

- 曖昧な内容は後々のトラブルにつながります。
 - 契約時に具体的な内容を確認しましょう。
-

⇒ 進捗状況をプロセスごとに明確化する

- プロジェクトを段階的に管理し、各フェーズでの進捗確認やフィードバックを計画的に行いましょう。
 - 明確なマイルストーンを設定し、進捗状況を視覚化して共有することで、問題を早期に発見できます。
-

⇒ 納品後のサポート内容を確認する

- 運用後の保守やトラブル対応についても事前に確認してください。
-

⇒ 認識のズレを防ぐための双方向コミュニケーション

- 双方が同じゴールや課題を共有できるよう、定期的にプロジェクト全体の目的や進行状況について話し合いましょう。
 - メールだけでなく、ビデオ会議や電話を活用して迅速に意思疎通を図りましょう。
-

13. よくある質問(FAQ)

Q: ヒアリングの際に何を準備すれば良いですか？

A: プロジェクトの背景や目的、解決したい課題、想定するユーザー像などを整理しておくスムーズです。簡単な手書きのアイデア図や参考サービスの例があるとさらに良いです。

Q: 必須の機能とオプション機能の違いをどう考えれば良いですか？

A: 必須機能は「プロジェクトの目的達成に不可欠なもの」、オプション機能は「あると便利だが後から追加も可能なもの」として区別してください。

Q: 予算内で対応できるか心配です。どうすれば良いですか？

A: 初回ヒアリングで予算感を伺い、優先度の高い機能や仕様を絞り込むことで調整が可能です。

Q: 技術的な専門知識がない場合でも対応してもらえますか？

A: もちろんです。専門用語や技術的な内容については、わかりやすく説明いたしますのでご安心ください。

Q: 保守やメンテナンスの契約は必須ですか？

A: 必須ではありませんが、システムの安定運用のため、定期メンテナンスをお勧めします。

Q: 小規模なプロジェクトでも相談できますか？

A: プロジェクトの規模に関わらず、お気軽にご相談ください。小規模でも最適な提案を行います。

14. GRILL のサポート内容



ヒアリング内容を基に
具体的な要件を一緒に整理します



プロジェクトの進捗を適切に管理し
トラブルを未然に防ぎます



運用開始後の質問や
不具合対応もご安心ください

15. お問い合わせはこちらから



<https://grill.bz/>