

グループ開発 について

グループ開発を行う目的

1. システム開発の全体イメージを把握

システム開発をどのように進めていくのか、基本的なウォーターフォールで理解します。

2. システム開発における基本スキルの習得

報告/連絡/相談、スケジュール作成、基本的なプログラミングスキル等を総合的に習得します。

3. 情報共有の重要性を理解

自身の事だけでなく、プロジェクト全体がどの位の進捗でどのような状況なのかを理解します。

グループ開発の目標

1. 指定した期間でのシステム完成

完成が最優先

なぜならば、完成しなければ顧客から二度と仕事に来なくなるからです。
しかし、完成しても不具合が多かったり要望との差異がある場合、未完成と同じです。
不具合が発生しないよう丁寧に作業し、十分なテストを行ってください。

2. スケジュール通りの作業

適当にスケジュールを組み、「出来ませんでした」では意味がありません。
作業量はそれぞれの力量を把握した上で組むようにしてください。
完成が難しい場合、担当講師にリスケジュールの相談をしてください。

開発の流れ

開発の流れは以下の通りです

1. プロジェクト発足
2. 要件定義・各種設計
3. 製造
4. テスト
5. リリース
6. 成果発表

プロジェクト発足

メンバーの役割決定や作成物の決定等、スタートさせる準備

プロジェクト発足

1. 開発システムの決定

● 自由作成

作成するシステムは自由に決定してください。
サンプルも用意しているので、サンプルを採用しても構いません。

● 開発規模

大きなシステムを作成する必要はありません。
必ずスケジュールと照らし合わせて吟味してください。

● 独自性

余裕がある場合はシステムに【他にはない独自性】を盛り込んでみましょう。

2. システム名の決定

● システム名

企業様にシステム名含め事前にお伝えします。
業務という意識を持って決めてください。

プロジェクト発足

3. プロジェクト名の決定

●プロジェクト名

システム名同様企業様にお伝えします。
業務意識を持って決定しましょう。

4. メンバーの役割を決定

●役割の概要と作業

メンバーそれぞれの役割を決定してください。
くじ引きやじゃんけんで決めるのではなく話し合いで決めましょう。
各役割のメンバーがその作業のみに全工数を割くのではなく、その成果物に対してどのように進めるかの決定権を持ちます。
作業はグループメンバーで協力して行いましょう。
次ページの表に各役割の詳細があります。

プロジェクト発足（役割表）

プロジェクトリーダー	話し合いのときにグループの最終的な意思決定を行います。 人員管理やスケジュール管理を行います。
サブリーダー	チーム作り、全体の意見の総意まとめを行います。ファシリテーターとしての役割も兼任し、1on1ミーティングなどチームの雰囲気作りを率先して行う役割です。
テクニカルリーダー	技術面の管理・サポートを行います。事前の技術調査や必要技術の選定、クラス構成と実装に関わる部分を受け持ちます。必要に応じてメンバー教育も含めます。
品質管理	成果物に対する責任を持ちます。納品物の把握、最新版の把握ドキュメント管理方法の策定を行います。
DB管理	データベースに関する責任を持ちます。実データの想定、機能に必要なSQLの構成組み立てを行います。

プロジェクト発足（具体的な作業例）

プロジェクトリーダー	各作業の落とし込み、詳細なスケジュール把握、人員の割り振り 設計、実装、テスト、その他システム作成に必要な作業
サブリーダー	打ち合わせ進行、各メンバーとの面談や意見の吸い上げ、各メンバーのサポートや折衝、設計、実装、テスト
テクニカルリーダー	作成するシステムの技術検証、技術策定、プロジェクト構成設計、 設計、実装、テスト、その他システム作成に必要な作業
品質管理	成果物の把握、ドキュメントの精査、ソースコードの把握、共有方法の策定、設計、実装、テスト、その他システム作成に必要な作業
DB管理	データベース設計、テーブル設計、実データの想定とテストデータの作成、マスターデータ投入、設計、実装、テスト、その他システム作成に必要な作業

プロジェクト発足

5. 概算スケジュール作成

おおまかなスケジュールを作成してください。

レビューのタイミングも考慮して時間に余裕のあるスケジューリングを目指しましょう。

いつ工程が終わる予定か、そこから逆算したらいつまでにレビューが必要か。

仕事においてスケジュール通りに行動することは非常に重要です。

スケジュール把握は全メンバー必ず行いましょう。

メンバーの誰に聞いても状況がわかることが理想です。

自分のタスクに責任をもち、誰に影響があるのかを把握しておいてください。

スケジュールに遅延等で修正が必要になった場合、メンバー全員でスケジュールの見直しを行いましょう。

要件定義・各種設計

作成するシステムの目的、必要な機能を明確にする

要件定義

1. システム概要

どのようなシステムを作成するか具体化していきます。

以下の観点で作成しましょう。

- 何を作成するか明確化されていること
- ターゲットユーザーの想定
- システムで何が提供できるか

また、以下の観点が盛り込まれているとより良いです。

- システムを導入することで得られるメリット
- 開発範囲

基本設計

1. 機能一覧

システム概要で洗い出した機能を細かい単位で分割しさらに具体化していきます。

例)

ユーザーメンテナンス（システム概要）→ユーザ検索、ユーザ登録（機能一覧）

以下の観点で作成しましょう。

- システムで提供できる機能が具体化されている
- 想定するターゲットユーザが要求しそうな機能か
- 各機能の開発優先順位がついている

2. 画面遷移

画面の流れを具体化しましょう。

画面内で機能が動作した後、どこの画面に遷移していくかを図で表します。

DB定義

1. テーブル設計

テーブルの構成を具体化していきます。

以下の観点で作成しましょう。

- 登録されるデータの具体的イメージ
- 機能が実現できるテーブル構成

データ型や制約は後でSQLを作成するまでに確定させておきましょう。

2. ER図

テーブル定義で上がったテーブルのER図にしていきます。

以下の観点で作成しましょう。

- すべてのテーブルが図に記載されている
- リレーションが正しくひかれている

詳細設計

1. 詳細設計

詳細設計は必須納品対象ではありません。

詳細設計はプロジェクトによって範囲が代わりますが、いずれにしても工数のかかる作業となります。

詳細設計は、誰が見ても実装を行えて且つ、システムに関わる人が認識を合わせることのできる設計書となります。

そのため、工数の関係で作成しない場合はその分、情報共有の比率を高めるべきであるということを留意しておきましょう。

製造

設計に従ってプログラミングを行う

1. プログラミング

誰がどの機能を担当するかはグループ内で話し合い決定しましょう。
決める際は個人の力量をしっかりと見極めて効率の良い分担ができると理想です。
また、自信がなくても積極的にプログラミングを行いましょう。
ドキュメント管理しか行わないでは意味がありません。

ここではプロジェクトメンバーとの連携が非常に大切です。
変数の名前、メソッドの引数・戻り値と自己判断で行うと結合できない事態に陥ります。
また、ソースコードのバージョン違い等でシステムが動かないなどが多くあります。
今までの事例では設計更新の連携不足、プログラム更新の連携不足、連絡の見逃し、等々で進捗遅れが非常に多くあります。

テスト

要件を満たしているか、設計が反映されているかチェック

テスト

1. テスト設計

テスト設計は必須納品対象ではありません。

作成した機能に対して、どのようなテストを行う必要があるかをまとめます。

以下に観点を一部記載しますが、グループでしっかり考慮して作成しましょう。

- 画面から入力する際のデータ型不正や桁数不正
- 設計した機能と動作がマッチしているか
- Web開発の場合は、URLや画面名

正常パターン、異常パターンの両方ができることが理想ですが、まずは正常パターンの完成を優先しましょう。

2. テスト実施

テスト実施の工数はしっかり取りましょう。

テストがされてない＝バグが残っている可能性が高いシステム になってしまいます。

また、テスト後のプログラム修正時間も考慮しましょう。

想定より修正に時間が取られてしまうことが多くあるので、余裕のあるスケジューリングにしましょう。

リリース

時間厳守、品質厳守

リリース

1. 納品物

発表会前日の 12:30 までに納品を完了させてください。

納品場所は講師の指示に従ってください。Gitにプッシュしただけでは納品完了とはなりませんので注意してください。

納品対象は以下となります。設計書類は形式は問いませんので必ずファイル化してください。

- システム概要
- 機能一覧
- 画面定義
- テーブル定義
- システムプログラム (warファイルかつJavaファイルも同梱)
- 初期データ (DDL、DML)
- 発表資料

発表資料に関しては発表会当日の 午前 9:00 までとします。

その他作成したドキュメントがあればすべて納品してください。

成果発表

時間厳守、品質厳守

成果発表

1. 発表資料

発表は資料を使って行います。以下の点を盛り込んだ資料にしましょう。

- 目次
- 作成物概要（どのようなシステムか）
- 何を目的としているか
- 概要スケジュール
- 設計時の機能一覧
- 実装できた機能一覧
- 設計と実装の差異とその要因
- どのくらいの時間・知識があれば設計時の機能がすべて作成できたか
- デモンストレーション
- 開発で得たこと
- 開発の反省と今後の目標
- 個人感想（各個人の感想ページに担当した機能を記載してください）

スケジュール例

実際のスケジュールを見てみよう

概算スケジュール例(8日間)



概算スケジュール例(10日間)



概算スケジュール例(11日間)



レビュー

顧客と方向性を認識合わせしよう

レビュー

1. レビュー

ウォーターフォール開発では各工程でレビュー発生します。

レビューは、開発者たちが想定しているものと顧客が想定しているものを擦り合わせる重要役割をになっています。レビュー対象のドキュメントは以下となります。

- システム概要
- 機能一覧
- 画面定義
- DB定義

各工程完了後、講師にレビューを依頼してレビューを行なってください。

ウォーターフォール開発では、基本的にレビューが完了したのちに次のステップへ進む流れとなります。

なぜならば、設計がのちの工程に与える影響が大きすぎるためです。

その点をしっかり意識しておきましょう。

レビューとドキュメント

1. レビュー申請

レビュー依頼は事前に講師に申請してください。

指定された時間に講師に対象ドキュメントの説明を行い、レビューを受けるようにしてください。

2. 対象ドキュメントの形式

必要な情報さえ有していればレビュー時の設計書等の記載方法は問いません。

必須納品の資料に関しては最終的に清書を行ってください。

レビューが完了したものは、その後変更はできません。

完成品（納品物）として相応しいものとしてレビューに出すよう心掛けましょう。

その他

開発において注意点をおさえておこう

その他

1. 進捗報告会

各グループの代表者一名ずつ講師が選出し、毎日16:30より進捗報告会を行います。
代表者は報告会の直前に指名しますので、いつでも誰でも報告できるようにしておきましょう。

2. 休憩

休憩タイミングは各グループで決めてください。

また、休憩する際の報告は不要ですが、外出（コンビニや自社の方と会う等）の際は報告してください。

3. グループ内定例

必須ではありませんが、業務をしているなかでミーティングはとても重要な情報共有手段です。
定められた時間でチーム内での問題点や情報を共有し、進むべき方向性を定めましょう。

よくある定例として、毎日の朝会・夕会などもありますので活用してみてください。

4. 質問

質問の制限はしませんが、できる限りグループ内で解決できるよう努力しましょう。

最後に

「作業効率を上げるにはどうすればよいか」

「全体の作業状況を把握するにはどうすればよいか」

「進捗が遅れないようにするにはどうすればよいか」

「進捗が遅延している人のフォローはどうか」

etc....

常に**グループ全体**で向上意識を持って取り組みます。

上記は実際の業務でも非常に重要です。このグループ開発が終わったら皆さん現場に配属されることでしょう。

その現場で出遅れないためにも精一杯努力して、各自が担当した役割を責任を持って完遂してください。

1人ひとりが本書を**熟読**し、理解したうえで作業に入りましょう。