

要件定義



要件定義プロセス:システムがどのように動作し、どのような機能を持つべきか(業務ニーズ(Needs))を具体的に決める段階
性能や利用方法も決める

利害関係者(管理者、ユーザ(利用者)、開発者など)がその内容を明確にして、合意することが重要

3種類の要件を定義する

1. **業務要件:**システムがビジネスの目標や目的を達成するために、どのような業務プロセスをサポートすべきかを決める
例:顧客管理の効率化、注文処理の迅速化
2. **機能要件:**システムが提供すべき具体的な機能、サービス、およびそれを実現するための画面やデータの流れを決める
例:顧客情報検索機能、業務機能間のデータの流れ
3. **非機能要件:**システムの性能や信頼性など、機能以外の品質面の要求を決める(応答時間など)
例:システムの監視サイクル、障害発生時の許容修復時間(「どれくらいの時間内に直してほしいか」という目安の時間)

要件定義プロセスでの注意点

- ① 要件定義プロセスで、システムに盛り込む(入れる)業務ルールを誤って解釈してしまうと、システムが意図しない動作をする原因となり、不適切なシステムが作られてしまう 例:授業管理システムで、15回の授業を14回と誤解して設計
- ② 要件定義が曖昧(はっきりしない)だと、開発途中で仕様変更や機能追加が起こりやすくなる
→スケジュールが遅れる →コストが増える
- ③ 要件定義が曖昧(はっきりしない)だと、適切な受け入れテスト(動作確認)ができない
※受け入れテストは、完成したシステムが要件定義の内容を満たしているかを確認するもの

要件定義



依頼者(クライアント(Client))



Communication, 合意(Agreement)



開発者(開発Team)

要件定義プロセス: システムがどのように動作し、どのような機能を持つべきか(業務ニーズ(Needs))を具体的に決める段階
性能や利用方法も決める

利害関係者(管理者、ユーザ(利用者)、開発者など)がその内容を明確にして、合意することが重要

3種類の要件を定義する

1. 業務要件: システムがビジネスの目標や目的を達成するために、どのような業務プロセスをサポートすべきかを決める

例: 顧客管理の効率化、注文処理の迅速化

2. 機能要件: システムが提供すべき具体的な機能、サービス、およびそれを実現するための画面やデータの流れを決める

例: 顧客情報検索機能、業務機能間のデータの流れ

3. 非機能要件: システムの性能や信頼性など、機能以外の品質面の要求を決める(応答時間など)

例: システムの監視サイクル、障害発生時の許容修復時間(「どれくらいの時間内に直してほしいか」という目安の時間)

要件定義プロセスでの注意点

① 要件定義プロセスで、システムに盛り込む(入れる)業務ルールを誤って解釈してしまうと、システムが意図しない動作をする原因となり、

不適切なシステムが作られてしまう 例: 授業管理システムで、15回の授業を14回と誤解して設計

② 要件定義が曖昧(はっきりしない)だと、開発途中で仕様変更や機能追加が起こりやすくなる

→スケジュールが遅れる →コストが増える

③ 要件定義が曖昧(はっきりしない)だと、適切な受け入れテスト(動作確認)ができない

※受け入れテストは、完成したシステムが要件定義の内容を満たしているかを確認するもの

要件定義



:システムがどのように動作し、どのような機能を持つべきか(業務ニーズ(Needs))を具体的に決める段階

性能や利用方法も決める

(管理者、ユーザ(利用者)、開発者など)がその内容を明確にして、合意することが重要

3種類の要件を定義する

1. :システムがビジネスの目標や目的を達成するために、どのような業務プロセスをサポートすべきかを決める

例:顧客管理の効率化、注文処理の迅速化

2. :システムが提供すべき具体的な機能、サービス、およびそれを実現するための画面やデータの流れを決める

例:顧客情報検索機能、業務機能間のデータの流れ

3. :システムの や など、機能以外の品質面の要求を決める(応答時間など)

例:システムの監視サイクル、障害発生時の許容修復時間(「どれくらいの時間内に直してほしいか」という目安の時間)

要件定義プロセスでの注意点

- ① 要件定義プロセスで、システムに盛り込む(入れる) を誤って解釈してしまうと、システムが意図しない動作をする原因となり、

不適切なシステムが作られてしまう 例:授業管理システムで、15回の授業を14回と誤解して設計

- ② 要件定義が曖昧(はっきりしない)だと、開発途中で や が起こりやすくなる

→スケジュールが遅れる →コストが増える

- ③ 要件定義が曖昧(はっきりしない)だと、適切な (動作確認)ができない

※受け入れテストは、完成したシステムが要件定義の内容を満たしているかを確認するもの

Requirements Definition



Requirements Definition Process: This is the stage where the specific **business needs** (i.e., how the system should operate and what **functions** it should have) are determined. It also includes decisions on performance and usage methods. It is crucial for stakeholders (administrators, users, developers, etc.) to clearly define and **agree** on these requirements.

Three types of requirements are defined:

1. **Business Requirements:** Determine the business processes that the system should support to achieve business goals and objectives.
e.g.: Streamlining customer management, speeding up order processing.
2. **Functional Requirements:** Define the specific functions, services, and the screens and data flows necessary to realize them.
e.g.: Customer information search function, data flow between business functions.
3. **Non-Functional Requirements:** Define the requirements related to quality aspects such as system performance and reliability (e.g., response time).
e.g.: System monitoring cycle, expected time to recover from failures.

Points of Caution in the Requirements Definition Process

1. If business rules to be incorporated into the system are misinterpreted during the requirements definition process, it can lead to unintended system behavior and result in the creation of an inappropriate system.
e.g.: Misinterpreting 15 lessons as 14 in a course management system design.
2. If the requirements definition is ambiguous, it can lead to frequent changes or additions to the specifications during development.
→ Delays in the schedule → Increased costs
3. If the requirements definition is ambiguous, it becomes difficult to conduct proper acceptance testing (to verify system operation). ※Acceptance testing is conducted to confirm whether the completed system meets the requirements defined in the requirements definition.