

「要求分析」と「機能要求/非機能要求」

九州地区審査委員 片山 徹郎
(宮崎大学 工学部 情報システム工学科)
<kat@cs.miyazaki-u.ac.jp>



このスライドの目的

- 「要求分析」と「機能要求/非機能要求」について語ります。
- 今年度は、モデル審査基準の追加課題として、「要求モデル」という項目があったためか、多くのチームが「要求分析」について、言及していました。



- 今回皆さんがETロボコンについて
 - どのような要求を見出し、
 - 分析をした結果、
 - 機能要求/非機能要求を抽出したのか

について、皆さんと一緒に見ていきたいと思います。



ETロボコン審査における「要求モデル」とは？

- ETロボコン審査における「要求モデル」
 - 今回の追加課題の審査対象の1つ
 - ちなみに、審査対象と審査内容は、
「ETロボコン モデル審査基準」
<http://www.etrobo.jp/2011/gaiyou/model.php>
に書いてあります。



- 「要求モデル」の審査内容を見てみましょう！



追加課題 – 要求モデル(抜粋)

- 製品やソフトウェアに対する**要求が、把握されているかどうか**を審査します。
具体的には、以下の内容を審査します。
 - 設計モデルで記述された**機能を導出するに至った経緯**(どのような**要求が存在し、どの要求をシステムで実現するかを決定したのか**)
が記述されていること
 - **機能的な要求や、品質、性能といった非機能的な要求が検討の過程およびその結果が**、記述されていること



モデリング実践トレーニング資料より

ETロボコン2011技術教育資料/ETロボコン実行委員会

組み込みソフトウェアの開発手順の一例

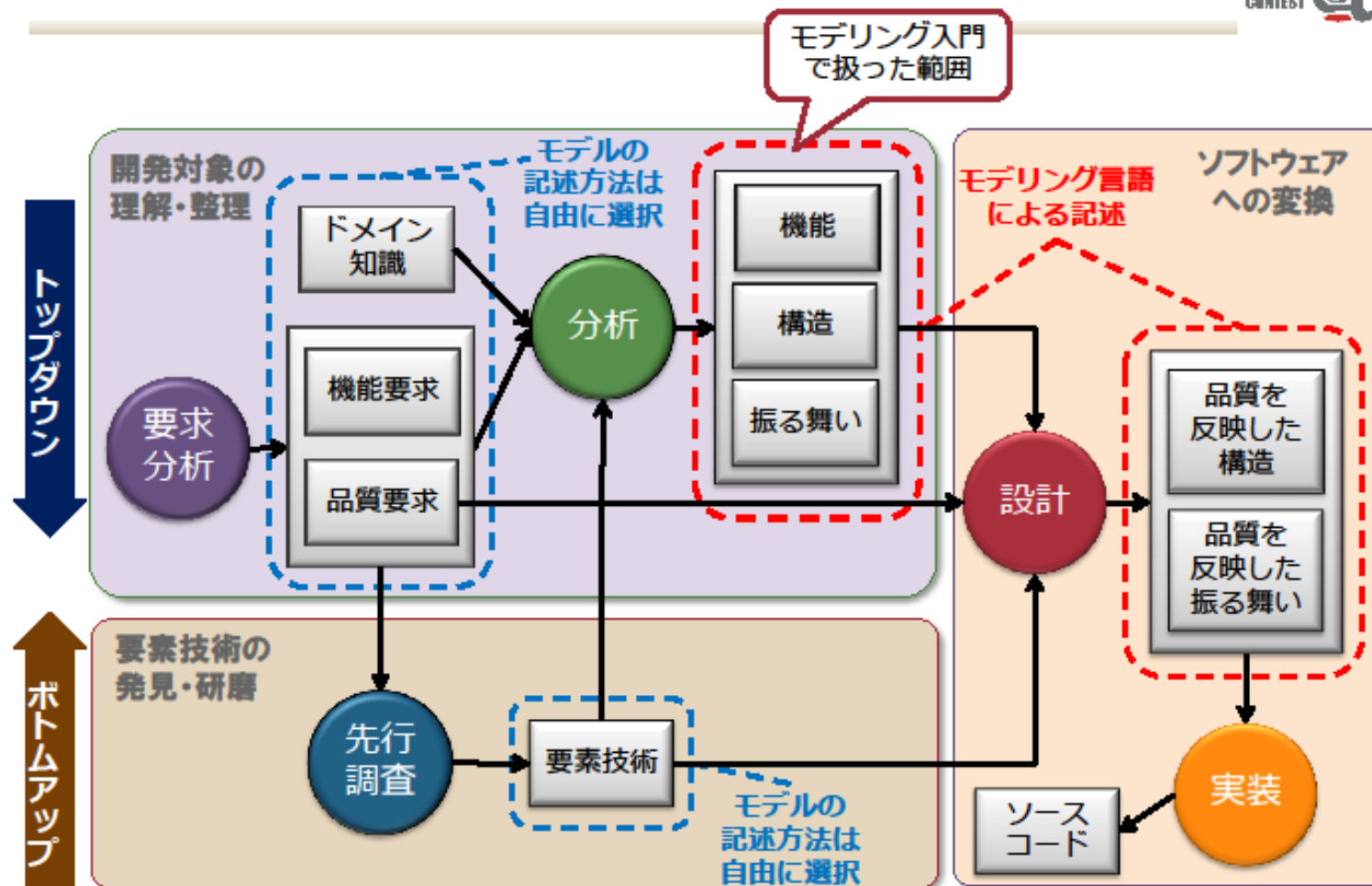


1. 製品への要求を明らかにする（**要求分析**）
 - 製品に対する要求を「機能」と「品質」の両面から把握する
2. 必要な知識・技術を明らかにする（**先行調査**）
 - 必要な「要素技術」を調査し、その実現可能性を判断する
3. ドメインを理解し、要求の実現方法を検討する（**分析**）
 - ドメインを理解した上で、製品として実現する「機能」を定義し、そのために必要な「構造」と「振る舞い」を明らかにする
4. ソフトウェアとしての実現方法を検討する（**設計・実装**）
 - 製品として実現する「品質」を定義し、そのために必要な「構造」と「振る舞い」を分析結果に反映させる
 - ソフトウェアとしての実現方法に変換する
 - 実現方法をソースコードに変換する

モデリング実践トレーニング資料より

ETロボコン2011技術教育資料/ETロボコン実行委員会

組込みソフトウェアの開発手順の一例



モデリング実践トレーニング資料より

ETロボコン2011技術教育資料/ETロボコン実行委員会

組込みソフトウェア開発とモデリング



- 実装につながるモデリングの中心となるのは、「分析」と「設計」
 - ここで作成されたモデルが、ソフトウェアのアーキテクチャへと変換される
 - したがって、ソースコードへの実装を前提としたモデリング言語・手法が使用される
- ただし、そこに至るまでの考え方をモデルに記述することで、モデルの理解性が格段に向上
 - 「要求分析」と「先行調査」
 - モデリング言語・手法は、考え方を伝えることができれば自由に選択出来る

モデリング実践トレーニング資料より

ETロボコン2011技術教育資料/ETロボコン実行委員会

要求分析



要求分析：製品への要求を明らかにする

■ 内容

- 製品に対する要求を、機能面と品質面の両面から把握する
 - 「難所をすべて突破してタイム部門で優勝を狙う」
 - 「無事完走を目指す」
 - 「環境変化に合わせた柔軟なチューニングが可能な作りにする」

■ モデリング

- モデルの冒頭で記載されると効果的
 - マインドマップ
 - ミスユースケース
 - 概念図（UMLのクラス図）
 - 図や自然言語による自由記述

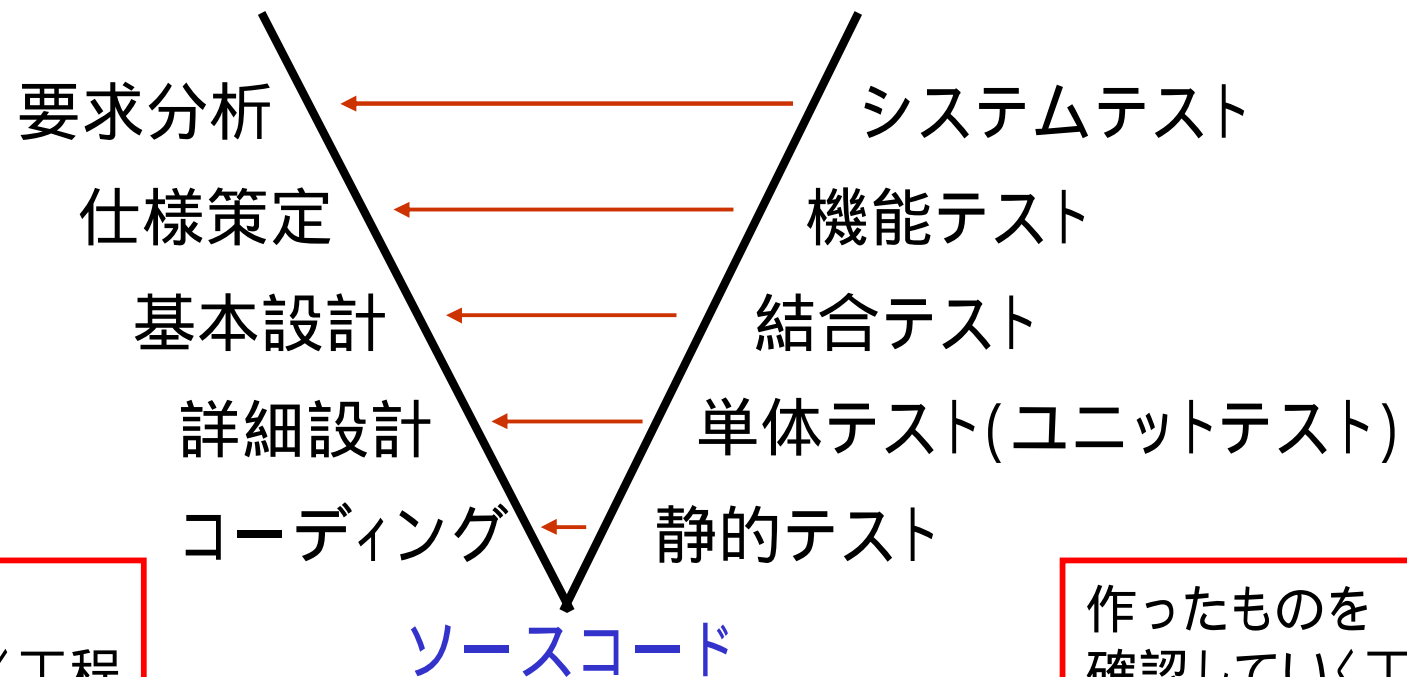
開発する目的や
製品のコンセプトが
明確になる！



ソフトウェア開発における「要求分析」とは？

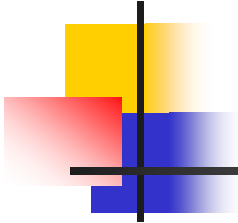
■ V字モデル

回帰テスト



ものを
作りこんでいく工程

作ったものを
確認していく工程



< 参考 > テストの工程(V字モデル)

- 静的テスト
 - バグになりやすいコーディングをしていないかチェックするコード監査や、複雑すぎないかを測定するメトリクス分析、レビュー、コードインスペクション、机上デバッグなどを行なう。
- 単体テスト
 - モジュール内のコードが正しいロジックかどうかをチェックし、予期しない例外についても確認する。
- 結合テスト
 - 関数の呼び出しなど、モジュール間のつながりが正しいかを確認する。
- 機能テスト
 - ソフトウェアに実装された機能が仕様書と合っているかどうかを確認する。
- システムテスト
 - ユーザが満足しているかどうかを確認する。
- 回帰テスト
 - 修正確認用のテストを実施する。



結局「要求分析」という工程は...

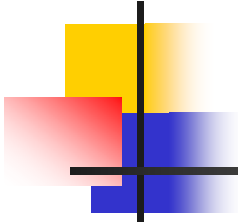


- 「要求分析」とは、一般に以下の作業がある。
 - 要求を聞きだす
 - 顧客やユーザとの対話によって、その要求を聞きだす。
 - 暗黙的だったり抽象的だったりする要求をどれだけ洗い出せるか
 - ユーザの抱えてる問題は何か、を見出す。
 - 要求を分析する
 - 粒度が一定ではない要求を明確化し、補い、具体化することによって、矛盾点や問題点を明らかにする。
 - QCD(Quality, Cost, Delivery)を考える。
 - この実現のために、例えば、マインドマップなどを使うと効果的！
 - 要求を文書化する
 - 要求を体系的に整理し、しっかり確定させ、設計のための土台とする。
 - 自然言語で記述する方法もあるが、ETロボコンでは、ユースケース図などのモデルを使う！

ソフトウェアの品質

- 実は、ソフトウェア品質の定義は様々！！！！
- 「品質とは誰かにとっての価値である」 by Gerald M. Weinberg
“Quality Software Management Vol.1: Systems Thinking”
 - つまり、品質とは、本来主観的なものであるという意識が重要
 - このソフトウェアは誰のために作っているのか？
 - ユーザにとっての価値とは何か？
- 品質を「目的への適合性」とする考え方
 - ISO/IEC9126が代表的





ソフトウェアの品質特性 (by ISO/IEC9126-1)

■ 機能性

- 合目的性
- 正確性
- 相互運用性
- 標準適合性
- セキュリティ

■ 信頼性

- 成熟性
- 障害許容性
- 回復性

■ 使用性

- 理解性
- 習得性
- 運用性

■ 効率性

- 時間効率性
- 資源効率性

■ 保守性

- 解析性
- 変更性
- 安定性
- 試験性

■ 移植性

- 環境適応性
- 設置性
- 規格適合性
- 置換性

■ <参考> 利用時の品質特性(ISO/IEC 9126-4)

- 有効性
- 生産性
- 安全性
- 満足性

では実際の提出物を見ていきましょう...

- コース分析

[[CKGロボコン部](#)] [[ちーくま](#)] [[SAGA組込ソフト研究会](#)] [[東海ど～かい!?](#)]

- ゴール指向分析

[[WSTJテクノ](#)] [[ロボサース](#)] [[隼人28号](#)] [[KTEC](#)]

- 多彩な要求

[[スーパーHTSくん](#)] [[はばたき隊](#)] [[HIBIKINO小次郎](#)] [[WPC](#)]

- 要求分析の前に...

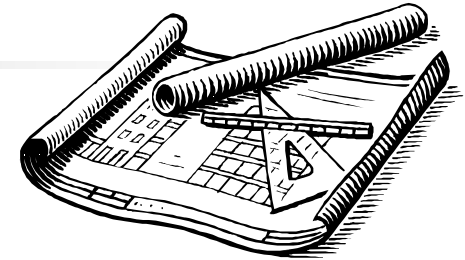
[[KFKF](#)] [[QNETモモちゃん](#)] [[ACOT](#)]

- 非機能要求のまとめにISO9126を利用

[[Nobu3SW\(R\)](#)] [[SDCバンビーズ](#)]



おわりに



■ 「要求分析」をしましょう！！

■ 要求を聞きだす

- 顧客やユーザとの対話によって、その要求を聞きだす。

- その本質は、問題発見です！
- ゴールありきで作ってませんか？
- 暗黙的な要求についても洗い出せてますか？

■ 要求を分析する

- 要求を必要に応じて明確化し、補い、矛盾点や問題点を明らかにする。

- 要求が漏れなく記載できてますか？
- 要求の全体像を俯瞰できてますか？

■ 要求を文書化する

- 要求を体系的に整理し、しっかり確定させ、システムの設計をするための土台とする。

- なぜこのユースケース図(設計モデルの機能)でOKなのかユーザが見て納得できますか？