

プロジェクトマネジメント支援ツール

平成27年度シラバス

2015年1月9日

国立情報学研究所

トップエスイープロジェクト

代表者 本位田 真一

1. 科目名

ソフトウェア・プロジェクトマネジメント支援ツール

(略称： プロジェクトマネジメント支援ツール)

2. 担当者

古宮 誠一

3. 本科目の概要

ツールを持たずにソフトウェア・プロジェクトマネジメントを行うことは、素手で猛獣と戦うようなものだという信念から、ソフトウェア・プロジェクトマネジメントに有効だと思われる多くのツールを開発してきた。本科目は、それらの中から特に有効と思われる、下記の5つツールについて紹介するとともに、どのようにすればそれらのツールを開発することができるのか、ツールの構築方法を明らかにする。

(1) インタビューによる要求抽出作業を誘導するシステム

インタビューによる要求抽出作業をインタビュー技術であると捉え、SE が行う顧客へのインタビュー作業をシステムが誘導するとともに、抽出した要求を基に要求仕様書を自動生成するシステムを開発した。このシステムを使用することにより、初心者 SE でも顧客要求を効率良く抽出できるだけでなく、抽出した要求を基に要求仕様書を自動生成することができる。

(2) プロトタイピング支援ツール

多くの顧客は、どのようなソフトウェアを開発して欲しいかということを SE に巧く伝えることが出来ない。その理由は、ソフトウェアを開発したこともなければ、類似のソフトウェアを使用したこともない顧客には、どのような事柄を SE に伝えれば SE に正しく理解してもらえるかということが判らないからである。この問題を解決するには、顧客が希望していると思われるシステムのプロトタイプ(実行可能なひな形)を開発して顧客に提示する方法(= ラピッド・プロトタイピング)が有効である。そこで、種々のラピッド・プロトタイピング支援ツールとその実現方法を紹介する。

(3) 要求仕様の内容の理解とその妥当性チェックを支援するツール

ソフトウェア開発作業が本格化する前に、顧客から提示された要求仕様の内容を理解するとともに、要求仕様の妥当性を確認することは非常に重要なことである。このため、要求仕様の内容の理解と妥当性確認のための作業を、経験豊富な人達がネットワーク上で議論することで行うことを考える。こうすることによって、要求仕様の不備(矛盾、当然ある筈の要求事項の漏れ、曖昧な点、疑問点など)を、ソフトウェア開発作業が本格化する前に検出し確認できるだけでなく、その時の議論の内容を上手に記録すると、それを読んだ人達が共通の理解をもって開発作業を行うことができるという大きなメリットを確保できる。そのためには、このときの議論の内容の記述漏れ

がなく、議論の経緯を含めて議論の内容を理解し易いように、議論の内容を構造化して即時記録することが求められる。そのような記録を可能にするためのモデルについて学習する。

(4) ソフトウェア分散協調開発支援環境

ソフトウェアの大規模化・複雑化に伴い、1つのソフトウェアを複数の人間が共同で開発することが多くなった。そこでは、メンバーの知識や経験を最大限に結集してソフトウェア設計作業を行えるように、設計課題に対する設計案の提案と、設計案の良し悪しとその根拠（これを **Design Rationale** と呼ぶ）を巡る議論を即時記録する仕組みが必要である。（**Design Rationale** の情報は設計上のノウハウと言うべきものである。しかし、**Design Rationale** の情報は、設計作業の担当者の脳裏に残るだけで、設計ドキュメントには一切記録されることはない。**Design Rationale** の情報を即時記録するシステム環境が実現すると、設計作業の担当者が、その脳裏にあることを積極的に吐き出すことになるので、プロジェクトマネージャは各担当者の検討の深さまでも把握できるので、究極のプロジェクトマネジメントが可能となる。）このような仕組みは、ソフトウェア設計作業をグループワークとして捉えることにより初めて可能となる。このことは、これまでソフトウェア設計作業を個人作業として捉え、設計方法論またはこれに基づく機械支援という形で支援してきたことと対照をなす。授業では、上記のようなソフトウェア開発を可能にするための開発環境とその実現方法を明らかにする。

(5) 自動プログラミングシステム

開発対象となるソフトウェアに求められる要求仕様を与えるだけで、要求仕様を満足するプログラム(原則として、そのソース・コード)を自動生成するシステムを自動プログラミングシステムと呼ぶ。自動プログラミングシステムの実現方法は、プログラム合成/生成技術から見て、①プログラム変換による方法、②演繹推論による方法、③帰納推論による方法、④部品合成による方法の4種類に分類できる。この授業では、これらがそれぞれどのような原理と方法で、プログラムの自動生成を可能にしているのかを明らかにする。これらの中で、最も実用性が高いのは部品合成による自動プログラミングシステムである。このため授業では、部品合成による自動プログラミングシステムの実現方法を特に詳細に明らかにする。

(6) ソフトウェア開発計画自動立案システム

どのようなライフサイクルモデルを採用しようとも、ソフトウェア開発プロジェクトには必ずそれを行うためのスケジュールがあり、各工程への開発要員の割り当てがある。スケジュール案から見ても要員割り当て案から見ても最適と思えるような開発計画(スケジュールと要員割り当て)案を自動立案することは誠に意義深い。これを可能にするツールは、初期計画の自動立案、工程遅延が発生したときに、それが後続工程に及ぼす影響の自動波及解析、納期を変えずに工程遅延を回復できるよ

うな計画案(存在するときにそれ)を自動立案する機能などを持つ。このようなツールは究極の工程管理支援ツールである。授業では、このようなツールの実現方法を明らかにする。

4. キーワード

インタビューによる要求抽出作業を誘導するシステム(A System to Navigate Interview-driven Software Requirements Elicitation Work), ラピッド・プロトタイプリング支援ツール(Rapid Prototyping Support Tool), 要求仕様の内容の理解と、その妥当性チェックを支援するシステム(A Support Tool to Understand the Contents of Software Requirements and to Check their Validity), 設計案に対する賛否の意見とその根拠を巡る議論(Design Rationale)を即時記録するシステム(Support Tools to Record Software Designs and their Design Rationales), ソフトウェア分散協調開発支援環境(An Environment to Support Software Distributed Collaborative Development), 自動プログラミングシステム(Automatic Programming Systems), ソフトウェア開発計画自動立案システム(An Automatic Schedule Planning System of Software Project)

5. 前提知識

特になし。

1. 講義計画

(1) 第1回 インタビューによる要求抽出作業を誘導するシステム

インタビューによる要求抽出作業を誘導するシステムとツールの構築方法について学習する。具体的な内容は次の通り。インタビューによる要求抽出作業に関する初心者 SE と熟練 SE の比較実験の内容。比較実験から判明した初心者 SE と熟練 SE の相違。要求抽出のために熟練 SE が用いていたインタビューのプロセス・パターン。熟練 SE が用いていたインタビューのプロセス・パターンと同様のプロセスを用いて初心者 SE を誘導する仕組みを構築する。そのために次のような仕組みを考える。インタビューの進捗管理テーブル。インタビューによる要求抽出作業を誘導するための2階層モデル。顧客回答を予測する方法。AND/OR 型の質問に対する誘導ルールの表現方法。誘導ルールに基づく顧客と SE のインターラクション。これらの仕組みを構築したツールの有効性を確認するための実験とその評価を明らかにする。

(2) 第2回 要求仕様の内容の理解と、その妥当性チェックするための支援ツール(その1)

ソフトウェア開発が本格化する前にやっておくべきこと、開発の対象となるシステムの分類、議論の内容を記録するためのモデルに求められる条件、要求仕様の理解と妥当性確認のための議論において議論の内容を構造化して即時記録するためのモデル、

『立場』というノードを設けずノード内記述項目とした理由、『要求項目』というノードを『部分要求項目』へ分解する過程、記述済みノードの記述内容を修正する方法、議論の参加者全員から賛成意見が得られていることの確認方法、各ノード内での記述項目詳細、議論の内容を記録するためのモデルに求められる諸条件とその充足方法一覧、ソフトウェア設計のための共通課題(酒屋の倉庫問題)、『酒屋の倉庫問題』を要求仕様として見た場合の疑問点、『酒屋の倉庫問題』を対象にした各ノードの記述例、

- (3) 第3回 要求仕様の内容の理解と、その妥当性チェックするための支援ツール(その2)
前回からの続きと演習。要求仕様の理解と妥当性確認のための議論の内容を構造化し、
即時記録するためのモデルに関する演習

- (4) 第4回 プロトタイピング支援ツールとツールの構築方法(その1)

プロトタイピング支援ツールとツールの構築方法。使用形態によるプロトタイプ
の分類。ツールがプロトタイピングに有効であるための条件。プロトタイプ作成技術
の分類。実行可能な仕様記述とその実現方法の分類。操作的アプローチとその実現方
法の分類・事例。ブラックボックス・アプローチとその実現方法の分類・事例。プロ
トタイピングでは不可能なこと。

- (5) 第5回 プロトタイピング支援ツールとツールの構築方法(その2)

前回からの続き。

- (6) 第6回 プロトタイピング支援ツールとツールの構築方法(その3)

前回からの続き。

- (7) 第7回 プロトタイピング支援ツールとツールの構築方法(その4)

前回の続き。

第4世代言語の分析とソフトウェア工学的評価。第4世代言語の定義と用途からの分析。
第4世代言語の機能モデル～その概念と必要性について～。ヒューマン・インタフェースから
の分析。第4世代言語の実現方式からの分析。プロトタイピング技術の分類と第4世代言語
の位置づけ。ソフトウェア開発支援ツールとしての第4世代言語の有効性評価。

- (8) 第8回 問題向き言語 POL(Problem Oriented Language)

プロトタイピング支援ツールとして位置付けることもできるストリームデータを扱う
プログラミング言語(グループ演習を含む)と第4世代言語について学習する。

- (9) 第9回 ソフトウェア分散協調開発支援環境(その1)

グループワークで行うソフトウェア設計作業のためのプロセスとその各プロセスへの支
援技術。Design Rationale の議論を即時記録し、再利用するシステムに求められる要件。
設計上の意思決定に十分な議論を尽くすことの必要性。gIBIS システムのモデルとその
利点・欠点。Potts のモデルとその特徴。議論の即時記録の必要性。部分課題ノードの
必要性。設計条件ノードの必要性。改良型 IBIS モデルの提案。改良型 IBIS モデルに
よる記述例。SIBYL/DRL による記述とその特徴。改良型 IBIS モデルで記録された設
計案とそれへのコメントの再利用の方法。事例ベース推論の利用による再利用性の拡大。

設計条件が他の設計案のそれと両立しない設計案の ATMS と DATMS による排除。

(10) 第10回 ソフトウェア分散協調開発支援環境(その2)

前回からの続きと演習。

(11) 第11回 自動プログラミングシステム(その1)

自動プログラミングシステムとは何か。実現技術からの自動プログラミングシステムの分類。プログラム変換によるプログラム合成の原理。プログラム変換によるプログラム合成システムの事例。命題論理と述語論理, 古典主義論理学と構成的数学, 導出原理, 定理証明によるプログラム合成の考え方, 定理証明によるプログラム合成システムの事例。

(12) 第12回 自動プログラミングシステム(その2)

定理証明によるプログラム合成システムの事例(前回の続き)。演繹推論と帰納推論, 類推(Analogy), 妥当な推論(Abduction), 仮説推論(hypothesis-based reasoning), 帰納推論のモデル, 枚挙(enumeration), 極限における同定(identification in the limit), 推論機械の構造, 言語の帰納推論, 要求仕様を事例で与えるという方式の難しさ, 例からの LISP プログラムの合成, モデル推論によるプログラム合成, プロダクションルールによるプログラム合成, 類推によるプログラム合成。

(13) 第13回 自動プログラミングシステム(その3)

これより部品合成による自動プログラミングシステムの説明に入る。手続き型言語のソースコードを部品化して登録し再利用するシステムの分類, 部品合成による自動プログラミングにおけるプログラムの生成過程, 部品の持ち方～再利用の対象からの分類, 自動プログラミングシステムと再利用支援システムにおける類似部品の扱い方の相違, プログラム部品に汎用性を持たせることが困難な理由, プログラム部品に汎用性を持たせる方法, 部品の検索方法, 部品の体系化とその方法, 部品のカスタマイズを容易にするための工夫, 部品の仕様記述に適した言語, 部品のソースコードをカスタマイズする方法とその部類, 部品同士の組み合わせと結合を容易にするための工夫, 部品同士を組み合わせるための戦略的知識, ソフトウェア再利用へのアプローチと再利用技術の分類, 自動プログラミングシステム PAPS が常備したソースコード部品の種類, 骨組み部品に部分部品を埋め込む方法, システムが常備するソースコード部品の比較,

(14) 第14回 自動プログラミングシステム(その4)

骨組み部品同士を組み合わせる方法, 部品を利用したプログラム開発支援ツールの実現戦略の比較, 部品合成によるプログラム合成を可能にするための知識と処理手順, 部品合成による自動構築方式, PAPS によるソフトウェア開発のプロセス

部品を利用したプログラミングシステムの比較, 部品合成による自動プログラミングシステムの2つの構ogram開発支援ツールの実現戦略の比較, 部品合成によるプログラム合成を可能にするための知識と処理手順, 要求仕様の与え方と理解の仕方。形式的な仕様記述による方法。(制限された)自然言語で与える方法。(階層化された)メニューの利用による方法, 部品合成による自動プログラミングシステムの比較, 部品合

成による自動プログラミングシステムの2つの構築方式, PAPS によるソフトウェア開発のプロセス

(15) 第15回 ソフトウェア開発計画自動立案システムとツールの構築方法

作業順序に関する制約。作業要員の割り当て条件に関する制約。作業要員の割り当て可能期間に関する制約。リソースの能力的限界に関する制約。2階層モデルと遺伝的アルゴリズムの導入によるソフトウェア開発計画自動立案。ソフトウェア開発計画立案のために調達が必要な要員のプロフィール自動生成機能。工程遅延の後続工程への影響波及解析。クラッシングによる工程遅延対策案の自動立案。休日出勤による工程遅延対策案の自動立案。ファストトラッキングによる工程遅延対策案の自動立案。3階層モデルの導入に基づく、クラッシング/休日出勤/ファストトラッキングの3方式混在による工程遅延対策案の自動立案。

7. 評価

出席日数とレポート課題への取り組み状況を総合して評価する。

8. 参考文献

- [1] 古宮誠一, “ソフトウェア開発に関する実態調査報告書,” 168 pages, 情報処理振興事業協会, March 1986.
- [2] 古宮誠一, “ソフトウェア再利用とその品質評価方法に関する調査報告書,” 152pages, 情報処理振興事業協会, March 1986.
- [3] 古宮誠一, “プログラム開発における知識ベースの応用と試行「プログラム自動合成システム PAPS の開発報告書」,” 352 pages, 情報処理振興事業協会, March 1988.
- [4] 古宮誠一, “プログラム開発支援技術と知識の応用ワークショップ報告書,” 525 pages, 情報処理振興事業協会, March 1988.
- [5] 古宮誠一, “ソフトウェア・プロトタイピング技術の調査研究「ワーキング委員会活動記録」,” (第1分冊 445pages) (第2分冊 470pages) (第3分冊 434pages) (第4分冊 405pages), 情報処理振興事業協会, March 1989.
- [6] 古宮誠一, “ソフトウェア開発における部品合成アプローチの有効性調査報告書,” 159 pages, 情報処理振興事業協会, March 1990.
- [7] 古宮誠一, “仕様記述実験結果報告書,” (上巻 598 pages) (下巻 255 pages), 情報処理振興事業協会, March 1990.
- [8] 古宮誠一, “(続)ソフトウェア・プロトタイピング技術の調査研究「ワーキング委員会活動記録」,” 情報処理振興事業協会, March 1991.
- [8] 古宮誠一, “ソフトウェア・プロトタイピング技術の調査研究ワーキング委員会活動報告書,” (上巻 546 pages) (下巻 640 pages), 情報処理振興事業協会, March 1991.
- [9] 古宮誠一, “ソフトウェア・プロトタイピング技術の調査研究ワーキング委員会活動報告書,” 374 pages, 情報処理振興事業協会, March 1991.

- [10] 古宮誠一, “ソフトウェア・プロトタイピング技術の調査研究「プロトタイピング支援ツール BPROT の開発報告書」, 296 pages, 情報処理振興事業協会, March 1991.
- [11] 古宮誠一, “ソフトウェアの信頼性に及ぼす人的要因の研究動向に関する調査報告書,” 130pages, 情報処理振興事業協会, March 1993.
- [12] 古宮誠一, “ソフトウェアの信頼性に及ぼす人的要因の調査研究成果報告書,” 202 pages, 情報処理振興事業協会, March 1994.
- [13] 古宮誠一, “データ中心設計過程支援のためのレポジトリ構築方法の実態調査,” 106 pages, 情報処理振興事業協会, March 1994.
- [14] 古宮誠一, “CASE ツールにおけるソフトウェア設計プロセスの実態調査,” 128 pages, 情報処理振興事業協会, March 1994.
- [15] 古宮誠一, “ソフトウェア設計過程の分析とモデル化の研究成果報告書,” 338pages, 情報処理振興事業協会, March 1994.
- [16] 古宮誠一, “合理的なソフトウェア開発プロセスの調査研究成果報告書（最終報告）,” 262 pages, March 1996.
- [17] 古宮誠一, “部品合成による自動プログラミング” 情報処理学会誌, Vol. 28, No. 10, pp. 1329-1345, Dec. 1987.
- [18] 古宮誠一, “事務処理ソフトウェア開発用簡易言語(第4世代言語)の現状と分析, 情報処理学会誌, Vol. 31, No. 7, pp. 1257-1269, Sep. 1990.
- [19] 古宮誠一, “高品質と高信頼性を確保する方法,” 人工知能学会誌, Vol. 6, No. 2, pp. 181-183, March 1991.