

ベイズ識別器による混乱予測に基づくソフトウェアプロジェクト 管理支援ツールの試作

近堂 高広, 安部 誠也, 水野 修, 菊野 亨
大阪大学 大学院情報科学研究科 情報システム工学専攻
E-mail: {t-kondou, s-abe, o-mizuno, kikuno}@ist.osaka-u.ac.jp

概要

ソフトウェア開発現場ではプロジェクト管理の重要性が益々高まってきている。特にプロジェクトが制御不能になる状態、いわゆる混乱状態を早期に発見することが求められる。本報告では、問題分析アンケートの作成、回収及び混乱予測までを一貫して Web 上で手軽に行えるツールを開発した。本ツールでは、プロジェクトマネージャーにアンケートを実施し、回答値と過去プロジェクトにおける混乱発生の有無との関係を調べ、進行中のプロジェクトが混乱に陥る確率をベイズ識別器を用いて予測する。さらに混乱要因を発見するために、各質問項目に対して混乱確率に関する感度分析を行う機能を実装した。ある企業の実際のデータを用いた適用実験を行い、ベイズ識別器に基づく予測や感度分析が有効であることを確認した。

A prototype of software project management tool based on runaway prediction using Bayesian Classifier

Takahiro Kondou, Seiya Abe, Osamu Mizuno, and Tohru Kikuno
Graduate School of Information Science and Technology, Osaka University
E-mail: {t-kondou, s-abe, o-mizuno, kikuno}@ist.osaka-u.ac.jp

Abstract

In software development project, necessity of project management has been increasing in recent years. It is thus required to discover runaway projects at an early stage. This paper describes development of a prototype of project management tool implemented as a Web application. The tool can handle tasks such as making questionnaire, collecting the answer, and predicting runaway probability. Project managers can fill out the questionnaire, examine the relation between the responses and the result of the past project, and predicts runaway probability by using Bayesian classifier. Furthermore, we implemented sensibility analysis function to identify runaway factors. The experiments using actual project data can conclude that the prediction by Bayesian classifier and sensibility analysis are useful for software project management.

1 まえがき

ソフトウェアの開発では、開発期間が次第に短くなる一方、高信頼性を求められるようになってきている。こ

うした状況下でソフトウェア開発プロジェクトが抱える問題点を早期に発見することの重要性が高まってきている。そのため、ソフトウェア開発プロジェクトにおける

リスクを分類して、その回避を行う技術の開発が望まれている。また、開発で余分な工数を費やしたくないという要求もあり、効率よくリスクを判断することが求められている。

ソフトウェア開発プロジェクトにおけるリスク分類は Boehm らによって古くから行われてきており [1], リスクを分析してプロジェクトの最終状態に関する予測を行う研究も行われている [6,9].

我々の研究グループでは、ソフトウェア開発プロジェクトの早期に行う問題分析アンケートによって、そのプロジェクトが最終的に混乱状態に陥るかどうかを判定する手法を提案してきた。過去に収集したアンケートのデータに対して統計的分析を行い、ロジスティック曲線に基づいてモデルを作成し混乱予測を行う手法 [7] や、アンケートのデータに対してクラスタ分析を適用し予測を行う手法 [5] も提案してきた。

これらの手法では、回答が得られないアンケート項目の場合にも、他の回答と同様に一つの回答値として取り扱っていた。しかし実際の開発現場での使用を考えた際に全てのアンケート項目に対して回答が得られるという前提は不自然で、更に、未回答のデータを適切に取り扱える手法は開発現場への導入の容易さにもつながる。したがって、このように回答が得られなかった項目を適切に取り扱うことが課題であった。さらに、これらの手法は回答を得てから予測モデルを作るまでの手順が煩雑であるため、進行中のプロジェクトに対する迅速な適用が難しいという問題もあった。

そうした問題点に対処するために、我々は未回答の項目の含まれるアンケートに対しても簡単に予測を行う手法として、ベイズ識別器を用いた手法を提案しており、本論文ではその手法を用いて実装したツールについて述べる。本ツールを用いることでアンケートの作成、実施、回収及びプロジェクトの混乱予測までを一貫して Web 上で手軽に行えるため、開発に余分な工数を費やさないようにしている。更に、回収したアンケートに対して感度分析を行う機能も実装し、リスク回避の支援を行った。

2 目的と準備

2.1 研究の背景

近年、ソフトウェアの規模が大きくなり、また開発期間が短くなるにつれ、プロジェクト管理の重要性が増している。

実際のソフトウェア開発現場では、稀にはあるがプロジェクトの進行状況すら誰にも把握できなくなってしまうようなことが発生する。このようなプロジェクトでは、多くの場合「混乱状態」と呼ばれる状態に陥ることを繰り返した後に、最悪の状態にまで発展してしまう。混乱状態に陥ったプロジェクトの全てが最悪の事態を招くわけではないが、こうした混乱状態を事前に予測できれば最悪の事態を防ぐことが出来る。

これまでソフトウェア開発プロジェクトの混乱状態を予測するためにいくつかの研究が行われてきた。これらの研究を通じて「開発現場のプロジェクトマネージャは無意識にはあるが、プロジェクトの混乱につながるリスク要因をつかんでいる」ということが分かってきた。

そこで我々はプロジェクトマネージャへのアンケート調査を実施し、アンケート調査に基づいて、プロジェクトの混乱とリスク要因との間に統計的モデルを作成し、そのモデルを適用して新規プロジェクトの混乱状態を予測する手法を提案してきた [5,7].

2.2 混乱プロジェクト

開発現場の状況が制御不可能になってしまうプロジェクトを混乱プロジェクトと呼ぶ。しかしこれでは漠然としているので、ここでは混乱プロジェクトを次のように定義する。

計画との相対誤差 混乱状態に陥ったプロジェクトは、工数・開発期間・納期などにおいて計画値からのずれが生じる。そこでこれらの計画値との相対誤差が基準値を超えているプロジェクト

開発プロセス 設計・レビュー・テストの各プロセスが組織の内部規約通り実施されていないプロジェクト

成果物の品質 プロジェクト成果物の品質が著しく悪いプロジェクト

これらの基準を基に、組織の SEPG(Software Engineering Process Group) により混乱状態に陥ったと認められ

たものを混乱プロジェクトと定義する (図 1 参照).

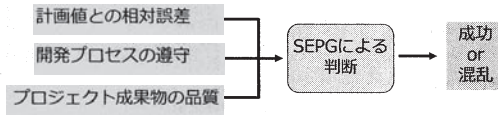


図 1: 混乱プロジェクトの評価

2.3 ベイズ識別器

ベイズ識別器はベイズの定理を利用してカテゴリカルデータをいくつかのクラスに高い精度で分類する手法として良く知られている。ベイズ識別器は各属性が互いに独立であるという仮定を置いているが、経験的に仮定が破られたデータセットであっても極めて精度の高い予測が可能であることが知られている [2]。この節では、ベイズの定理とベイズ識別器の概要について説明する。

2.3.1 ベイズの定理

ベイズの定理は、事前確率を事後確率に変換するものである [3]。

確率変数 A, B において、事前確率 $P(A), P(B)$ 、事後確率 $P(A|B)$ とするとき、次のような関係が成り立つことが知られている。

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

2.3.2 ベイズ識別器

Q_1, Q_2, \dots, Q_n を属性集合とし、 C を予測を行うクラスとする。ここでは、属性集合、クラス共に名義尺度として扱う。

属性集合が、 $Q_1 = q_1, \dots, Q_n = q_n$ と与えられたとき、クラス $C = runaway$ となる確率 $P(C = runaway | Q_1 = q_1 \wedge Q_2 = q_2 \wedge \dots \wedge Q_n = q_n)$ は、ベイズの定理を用いて次のように表される。

$$\frac{P(Q_1 = q_1 \wedge \dots \wedge Q_n = q_n | C = runaway)}{P(Q_1 = q_1 \wedge \dots \wedge Q_n = q_n)} \times P(C = runaway)$$

$P(C = runaway)$ は学習データセットから簡単に求めることができる。また、 $P(Q_1 = q_1 \wedge \dots \wedge Q_n = q_n)$

はクラス C の値に関わらず一定の値であり事後確率の合計が 1 になるための正規化係数と考えることができる。そこで、 $P(Q_1 = q_1 \wedge \dots \wedge Q_n = q_n | C = runaway)$ を求めることに焦点を当てる。

ベイズの定理を再び適用すると、この部分は次のように変形できる。

$$P(Q_1 = q_1 | Q_2 = q_2 \wedge \dots \wedge Q_n = q_n, C = runaway) \\ \times P(Q_2 = q_2 \wedge \dots \wedge Q_n = q_n | C = runaway)$$

この式の第 2 項目も同様にして、

$$P(Q_2 = q_2 | Q_3 = q_3 \wedge \dots \wedge Q_n = q_n, C = runaway) \\ \times P(Q_3 = q_3 \wedge \dots \wedge Q_n = q_n | C = runaway)$$

ここで、各属性 $Q_i (1 \leq i \leq n)$ が互いに独立であると仮定すると、

$$P(Q_1 = q_1 | Q_2 = q_2 \wedge \dots \wedge Q_n = q_n, C = runaway) \\ = P(Q_1 = q_1 | C = runaway)$$

が成り立つ。

よって、 $P(Q_1 = q_1 \wedge \dots \wedge Q_n = q_n | C = runaway)$ は次のようになる。

$$\prod_{i=1}^n P(Q_i = q_i | C = runaway)$$

以上より、ベイズ識別器により、クラス $C = runaway$ となる確率は次の式で求められる。

$$P(C = runaway | Q_1 = q_1 \wedge \dots \wedge Q_n = q_n) \\ = \frac{(\prod_{i=1}^n P(Q_i = q_i | C = runaway)) \times P(C = runaway)}{P(Q_1 = q_1 \wedge \dots \wedge Q_n = q_n)}$$

全てのクラスについて同様に確率を計算し、最も確率が大きいクラスに分類することで予測を行うことが出来る。

3 ベイズ識別器に基づく混乱予測ツール

本研究で提案する混乱予測ツールは図 2 に示すように、問題分析アンケートの質問項目作成、アンケートの回収、プロジェクト成否の入力、プロジェクトの混乱予測、及び混乱確率に関する感度分析までを Web 上で一貫して行う。

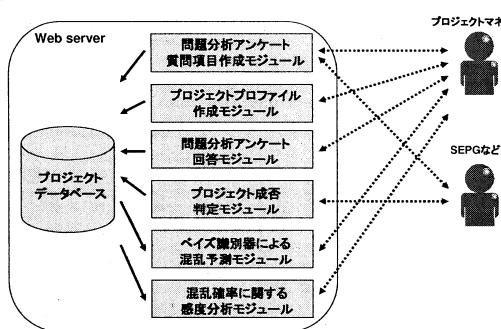


図 2: 混乱予測ツール

図 3 は本ツールを使用した新規プロジェクトの混乱予測までの手順を示している。また、新規プロジェクトの混乱確率が高いと予測された場合には、混乱要因の発見を支援するための感度分析を行うことが出来る。以降では新規プロジェクト混乱予測までの流れと、感度分析機能について説明する。

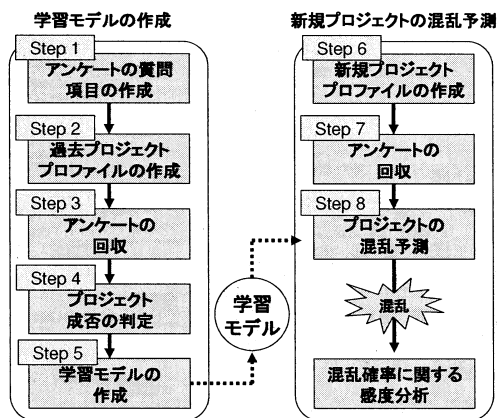


図 3: 混乱予測ツール使用の流れ

3.1 ツールによる混乱予測までの流れ

最終的にベイズ識別器によるプロジェクト混乱予測の結果を得るためには、過去プロジェクトのアンケートデータから成功・混乱について学習する必要がある。ここでは過去プロジェクトのアンケートデータから得られ

る学習モデルの作成、及び、新規プロジェクトの混乱予測について説明する。

3.1.1 学習モデルの作成

- Step 1: 質問項目の作成
SEPG 又はプロジェクトマネージャが、プロジェクトが抱える問題を抽出するために実施するアンケートの質問項目を作成する。アンケートの質問項目は組織の内部規約やプロジェクト特性を考慮して作成する必要がある。また各質問項目に対する多段階評価の回答値も設定する。
- Step 2: 過去プロジェクトプロフィールの作成
プロジェクトマネージャが、プロジェクト名、内容、規模、開始時期等のプロジェクトの情報を入力する。これらの情報は個々のプロジェクトを判別する目的で入力し、混乱予測には使用しない。
- Step 3: アンケートの回収
Step 2 で入力した過去プロジェクトに関して、Step 1 で作成したアンケートにプロジェクトマネージャが回答を行う。
- Step 4: プロジェクト成否の判定
Step 2 で入力したプロジェクトに関して、SEPG が 2.2 節で述べた混乱プロジェクトの定義に基づき、混乱あるいは成功を判定する。
- Step 5: 学習モデルの作成
Step 3 で得られたアンケートの回答と Step 4 で判定されたプロジェクト成否からベイズ識別器による学習モデルを作成する。このとき、アンケートの回答に未回答の項目があるものも問題なく扱える。

3.1.2 新規プロジェクトの混乱予測

- Step 6: 新規プロジェクトプロフィールの作成
Step 2 と同様に、プロジェクトマネージャが進行中の新規プロジェクトの情報を入力する。
- Step 7: アンケートの回収
Step 6 で入力した新規プロジェクトに関して、プロジェクトマネージャがアンケートの回答を行う。

• Step 8: プロジェクトの混乱予測

Step 7 で回答したアンケートデータに対して、Step 5 で得られた学習モデルを用いて、混乱プロジェクトとなる確率及び成功プロジェクトとなる確率を計算し、確率が高い方のクラスに分類することで混乱予測を行う。

		回答値				
		1	2	3	4	5
質問項目	Q_1	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}
	Q_2	P_{21}	P_{22}	P_{23}	P_{24}	P_{25}

	Q_n	P_{n1}	P_{n2}	P_{n3}	P_{n4}	P_{n5}

図 5: 感度分析結果

3.2 混乱確率に関する感度分析

バイズ識別器の特徴の1つとして、質問項目をそれぞれ独立して扱ってモデルを作成するという点が挙げられる。その特徴を活かして、進行中の新規プロジェクトのある質問項目の回答値によって、どのように混乱確率が変わるかを確かめられる感度分析機能を実装した。感度分析とは関係する1つの変数が変動したとき、結果がどのように変化するかを検討する手法であり、その変数の結果への影響度合いを定量的に計測できる。

図 4 はある質問項目 Q_i における回答値 q_i を変動させたときの混乱確率に関する感度分析の結果を表わしている。この質問項目に対する回答値が5段階である場合、他の質問項目の回答値は固定したまま、 Q_i だけが1~5(5段階)まで変化したときの混乱確率を求める。

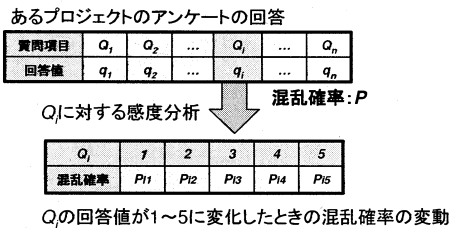


図 4: ある質問項目 Q_i に対する感度分析

図 5 はすべての質問項目に対して感度分析を行った結果を示している。質問項目数 n のそれぞれの回答値が5段階である場合、混乱確率の変動は $n \times 5$ 行列の形で表される。この感度分析機能により、進行中のプロジェクトが混乱と予測された際に、どの質問項目の示すリスクを下げれば、どの程度混乱確率が下がるかがわかる。したがって、混乱を回避するために注意すべき質問項目をプロジェクトマネージャに提示できるため、効率的なプロジェクト管理につながると期待される。

3.3 ツールの実装環境

混乱予測ツールは Web 上で動作するアプリケーションとして実装し、スクリプト言語である Perl を用いて記述した。Perl の開発規模はほぼ 2200 行になり、画面遷移ページの総数は 20 ページとなった。またプロジェクトデータベースの管理は MySQL [10] を使用し、バイズ識別器による混乱予測の計算は統計解析ソフト R [11] を使用した。

4 適用実験

本ツールを用いて、バイズ識別器に基づいたプロジェクト混乱予測の有効性を確認する実験を行った。また、混乱要因の発見につながる感度分析に関する実験も行った。ここでは 2 種類の実験についてそれぞれ説明する。

4.1 対象プロジェクト

以降の 2 つの適用実験では、実際に行われたソフトウェア開発プロジェクトから得られたアンケートデータを使用して行う。これらのプロジェクトは組み込みソフトウェア開発プロジェクトであり、実際のソフトウェア開発企業の協力を得て、1996 年から 1998 年までに開発された 40 プロジェクトのマネージャから回答を得た。

また、これらのプロジェクトはすでに開発が終わっているため、企業の SEPG により 2.2 節で述べた基準を考慮して混乱プロジェクトか否かの成否の判定はなされている。

4.2 問題分析アンケート

問題分析アンケートはリスク要因に関する論文 [1,4,8] と協力企業の内部規約を調査して作成した。その結果、

表1に示すように、要求内容に関する問題点、見積もりに関する問題点、プロジェクト体制に関する問題点、工程計画に関する問題点、問題発生時の対応に関する問題点、の5つのカテゴリに整理された。これらのカテゴリには複数の質問項目が設置されており、回答は4段階評価で行った。

表1: 問題分析アンケート

1. 要求内容に関する問題点	
QR1	ソフトによる実現を要求する側が、何を要求したいか分かっていなかった
QR2	要求側の説明力不足
QR3	実現側の理解力不足
QR4	実現側の理解内容・構想を、要求側に説明・確認不足
2. 見積もりに関する問題点	
QE1	例外処理などの見積もり項目抜け (見えている範囲しか見積もっていなかった)
QE2	見積もりの大切さの認識不足 (安易に約束してしまった)
QE3	政治的圧力に妥協してしまった
QE4	技術的課題を過小評価してしまった
QE5	過去の成功パターンで安易に見積もった (見積もり根拠が不明確であった)
3. プロジェクト体制に関する問題点	
QO1	要求スキルを考慮せず、その時に当てられる人員しか確保しなかった
QO2	プロジェクトの開始時期に必要な人員が確保できなくても、最初のうちだからなんとかなると思いき手を打たなかった。
4. 工程計画に関する問題点	
QP1	作業分割構造 (WBS) およびプロジェクト組織構造の明確化不十分
QP2	各作業分担の工程毎成果物の定義が不十分
QP3	マイルストーン、チェックポイント、レビュー時期の設定がなかった
QP4	計画に対する設計者全員のコミットメントがなかった (担当者は実現不可能と思っていた)
QP5	マネージャによる実現性の検証不足
QP6	進捗管理方法不明確
5. 問題発生時の対応に関する問題点	
QM1	問題状況をマネージャへ適切に報告できなかった
QM2	問題報告するもマネージャが適切に対処しなかった
QM3	状況を整理し上流工程の再評価をしなかった
QM4	人員投入のみなど量的打ち手のみになってしまった
QM5	対策に対する見通しが不足
QM6	応援を入れる時期が遅かった

4.3 実験1: プロジェクトの混乱予測

4.3.1 実験の概要

実験で対象とする40プロジェクトはすべて過去のプロジェクトであり、新規プロジェクトのアンケートデータは存在しない。そこで、本ツールを使用してジャックナイフ法による混乱予測実験を行った。ジャックナイフ

法では1つのプロジェクトを予測データとして取り出し、残りのプロジェクトのデータを学習データとして、取り出した1つのデータの予測を行う。これを40回繰り返すことによりすべてのプロジェクトのデータを1回ずつ予測し、その精度を評価する手法である。ここでは取り出した1つのプロジェクトを新規プロジェクトと見なして実験を行った。

4.3.2 実験結果

ジャックナイフ法により、得られた予測結果とSEPGによって判定された実際の結果を比較した実験の結果を表2に示す。ここでは、ベイズ識別器による予測結果において混乱確率が50%以上のものを混乱、50%未満のものを成功のクラスに分類した。

表2に示すように、40プロジェクト中33プロジェクトを正しく予測することができ、予測精度は0.825となった。また、再現率と適合率から求められるF値も比較的高い値となることがわかった。このことからアンケートの回答を用いたベイズ識別器による混乱予測が可能であることが確認できた。

表2: プロジェクト混乱予測の結果

		予測		予測精度	0.825
		成功	混乱		
実測	成功	22	5	再現率	0.846
	混乱	2	11	適合率	0.688
				F値	0.759

4.4 実験2: 感度分析による混乱要因の発見

4.4.1 実験の概要

実験2では感度分析機能がどの程度有効なのかを検証する。対象とするプロジェクトは、混乱状態に陥り、かつツールが混乱と予測したものである。表1のアンケートの回答値は、1(該当しない)、2(ほとんど該当しない)、3(やや該当する)、4(該当する)のいずれかになっている。したがって回答値が1に近いほど成功しやすいと考えるのが自然である。従って、実際に混乱したプロジェクトに対して、得られた回答値を1段階小さくするだけで混乱確率が50%未満になると予測できれば非常

に有難い。条件に該当するのは40個中11個のデータとなり、それらを対象に分析を行った。

4.4.2 実験結果

11個のデータに対して感度分析機能を実行した結果は表3のようになった。表3は、プロジェクトNo、変更したマトリクスの項目と値、混乱確率の変動を示している。

感度分析を行った結果、11個のプロジェクトのうち7個のプロジェクトの混乱確率を下げることに成功した。変更すると混乱確率を小さくできるマトリクスは主に、QR(要求内容に関する問題点)とQE(見積もりに関する問題点)となった。特にQR2(要求側の説明力不足)は頻出しており、混乱に大きく影響を与えるマトリクスであると考えられる。この企業の場合、ソフトウェアプロジェクトを進める上で、より綿密に顧客と打ち合わせを行うべきであろう。

表3: 実験2の結果

プロジェクト No	変更した マトリクス	混乱確率(%)	
		変更前	変更後
10	QR1:3 → 2	97	27
	QR2:3 → 2		0
11	QR2:3 → 2	100	0
	QR4:3 → 2		0
13	QR2:3 → 2	96	0
	QE2:4 → 3		41
	QM3:3 → 2		0
14	QR2:3 → 2	71	0
	QR4:3 → 2		24
	QE1:3 → 2		20
	QE4:4 → 3		38
15	該当なし	100	なし
16	QR2:3 → 2	99	0
25	QR2:3 → 2	99	0
30	該当なし	99	なし
31	該当なし	100	なし
38	該当なし	99	なし
40	QR2:3 → 2	99	0

5 まとめと今後の課題

本研究ではベイズ識別器による混乱予測に基づくソフトウェアプロジェクト管理支援ツールの試作を行った。アンケートの作成から混乱予測までを行うことができ、プロセス管理に大きく役立つと考えられる。実験を通して感度分析機能の有効性を調べた結果、プロジェクトマネージャーに改良点の提示を行うことも可能であるとわかった。

今後の課題としては実際にプロジェクトマネージャーにツールを使用してもらい、その有効性を検証することが挙げられる。また本実験では混乱予測をベイズ識別器に限定したが、それに限らず他の予測手法を組み込み、ユーザが自由に選択できるようにすることも考えられる。取得するプロジェクトの情報を増やし、工数やバグ数の予測を自動化することやデータベースに蓄積されているデータを有効に活用することも挙げられる。現状はまだ試作段階であるためこれから機能を増やしていくと考えている。

参考文献

- [1] B. W. Boehm. Industrial software metrics top 10 list. *IEEE Software*, 4(5):84–85, 1987.
- [2] P. Domingos and M. J. Pazzani. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 29(2-3):103–130, 1997.
- [3] R. O. Dura, P. E. Hart, Daviid G., 尾上守夫監訳. パターン識別. 新技術コミュニケーションズ, 2001.
- [4] R. Fairley and P. Rook. Risk management for software development. In *Software Engineering*, pp. 387–400. IEEE CS Press, 1997.
- [5] T. Hamasaki, O. Mizuno, T. Kikuno, and Y. Takagi. Cluster analysis of questionnaire for risk management and prediction of confused projects. In *Proc. of Software Symposium 2002*, pp. 159–166, 2002. In Japanese.
- [6] J. Jiang and G. Klein. Software development risks to project effectiveness. *Journal of Systems and Software*, 52:3–10, 2000.

- [7] O. Mizuno, T. Kikuno, Y. Takagi, and K. Sakamoto. Characterization of risky projects based on project managers' evaluation. In *Proc. of 22nd International Conference on Software Engineering*, pp. 387–395, 2000.
- [8] R. C. Williams, G. J. Pandelios, and S. G. Behrens. Software risk evaluation (SRE) method description (version 2.0). Technical Report CMU/SEI-99-TR-029, Software Engineering Institute, 1999.
- [9] C. Wohlin and A. A. Andrews. Prioritizing and assessing software project success factors and project characteristics using subjective data. *Empirical Software Engineering*, 8:285–303, 2003.
- [10] 田中ナルミ, 阿部忠光. 標準 MySQL. SoftBank Publishing, Tokyo, 2005.
- [11] 間瀬茂, 神保雅一, 鎌倉稔成, 金藤浩司. 工学のためのデータサイエンス入門 フリーな統計環境 R を用いたデータ解析. 数理工学社, Tokyo, 2004.