

# アンケート回答方式によるソフトウェアプロジェクトの混乱予測 — 因子分析によるアンケート項目の集約 —

水野 修†  
Osamu Mizuno

高木 徳生‡  
Yasunari Takagi

菊野 亨†  
Tohru Kikuno

表 1: アンケート調査表

## 1. まえがき

ソフトウェア開発プロジェクトでは多くのリスク要因が複雑に関係して問題が発生することが知られている [1]。実際に、プロジェクトの状況が把握できなくなって、プロジェクト管理が混乱することが少なくない [4]。そのため、プロジェクトの混乱を予測する手法の開発が強く求められているが、人間の知的作業を相手にするのでその開発はきわめて難しい。本研究ではアンケートから得られるデータを統計的に処理して、プロジェクトの混乱予測に利用することを目指す。

## 2. 背景 (混乱予測)

### 2.1 アンケートに基づく混乱予測

これまでに、ソフトウェアプロジェクトの混乱を統計的に予測する研究を行ってきた [2]。ここではリスク要因のアンケート結果から、プロジェクトの混乱に最も強く関係のある要因を発見し、それに基づいたプロジェクトの混乱予測をしていた (図 1 参照)。



図 1: 混乱予測手順

### 2.2 アンケート項目 [2]

プロジェクトマネージャに対して表 1 に示すアンケート調査表を配布し、彼らの手がけたプロジェクトに対する回答を回収する。アンケート表は経験的に作成されており、5 個の大項目とその中の 23 個の小項目から構成される。これらの項目に対して、「はい」、「いいえ」、「どちらでもない」、「分からない」から回答する。このアンケートは、過去の文献におけるリスク分類を基に協力企業における開発者の経験を加味したものとした。

### 2.3 モデル式 [2]

文献 [2] で作成されたプロジェクトの混乱予測モデルは次のようになっている。

$$P(y|x_2, x_4) = \frac{e^{-5.25+2.72x_2+3.98x_4}}{1 + e^{-5.25+2.72x_2+3.98x_4}}$$

ここで、 $P(y|x_2, x_4)$  は事象  $y$ (混乱) が発生する条件付確率を表す。このモデルでは 5 つの大項目からステップワイズ法で 2 つの説明変数 ( $x_2$ : 見積,  $x_4$ : 計画) を採用している。

### 2.4 問題点

文献 [2] でのモデル化の手法では、2 つの変数  $x_2, x_4$  のみを利用しており、結果的に多くの項目の影響を排除していた。開発現場への導入の上ではその点が問題になっていた。本研究ではそれを克服する試みについて述べる。

†大阪大学 大学院情報科学研究科

‡オムロン株式会社 ソーシャルシステムズビジネスカンパニー

1. 要求仕様
1.1 要件が不明確なままでの要求
1.2 要件の引出し不足
1.3 要件の理解力不足
1.4 要件に対する顧客側、実現側相互の合意不足
2. 見積り
2.1 見積り項目不足
2.2 見積りの重要さの認識不足
2.3 非技術的圧力に妥協
2.4 技術的課題に対する楽観的見積り
2.5 見積りの根拠不足
3. プロジェクト体制
3.1 要員のスキル不足
3.2 プロジェクト体制整備の必要性認識不足
4. 工程計画
4.1 作業に対する責任分担不明確
4.2 作業成果物定義不十分
4.3 マイルストーンやレビュー時期の設定不足
4.4 計画に対する関係者全員のコミットメントなし
4.5 計画に対する上級マネージャのレビュー不足
4.6 進捗管理方法不明確
5. 進捗管理
5.1 技術的側面でのリスク管理不足
5.2 モラル不足
5.3 工数不足
5.4 要件、仕様変更管理不足
5.5 進捗状況報告不足
5.6 進捗管理データ不足

## 3. 新しいアプローチ

### 3.1 適用手順

提案する新しいアプローチは次の 3 つのステップから構成されている。なお、Step 2 が文献 [2] での手法にはなかった新しい操作である。

**Step1** アンケート項目の列挙 まず、アンケート項目を列挙する。今回は表 1 に示すものを利用した。

**Step2** アンケート項目の集約 次に、因子分析を用いて、アンケート結果を幾つかの特徴的な要因へと集約させる。

**Step3** モデル式の作成 最後に集約された因子を説明変数とするロジスティック回帰モデルを作成する。

### 3.2 Step2 (因子分析)

本来、様々なデータは互いに完全に独立ではなく、関連のあるデータも存在している。因子分析により、それらのデータから、独立した因子を探り出し、収集したデータを有効に用いたモデルを作成する。

因子分析の基本モデルは次式で表せる [3]。

$$Z = FA' + UD$$

ここで  $Z$  がアンケートに対する回答データにあたる。また、 $A$  は因子負荷行列と呼ばれる。因子分析の結果、アンケート項目は幾つかの因子に集約される。集約された因子はアンケート項目を大局的に見たりリスク要因を表す。更に、因子負荷量の傾向に基づき適切な名前付けを行うことで、リスク要因の抽出が行える。

最後に  $F$  は因子スコア行列と呼ばれる。因子分析の結果を外的要因と関連付けて考える場合に利用される。本研究の場合、プロジェクトがそれぞれの因子に対してどれだけ高い危険性を持つのかを  $F$  が表している。

### 3.3 Step 3 (ロジスティック回帰)

Step 2 で求めた因子を説明変数とするロジスティック回帰分析を行い、混乱予測のためのモデル式を得る。

## 4. 実データへの適用

### 4.1 対象プロジェクト

本報告で対象としたプロジェクトは、ある企業において1996年から1998年に実施された37個の組み込みソフトウェアの開発プロジェクトである。これらに対してリスク調査アンケートを配布し、回答を得た。

### 4.2 Step 2 の結果 (要因の抽出)

まず各プロジェクトに対するアンケートの回答値に因子分析を適用した。因子数の決定には75%分散法を、因子の抽出法には主因子法を、また因子解の回転法としてはバリマックス変換を利用した。

因子分析を実行した結果、アンケート項目は7つの因子 (Fac.1 ~ Fac.7) に集約された。(表2, 表3参照)

表2: アンケート項目に対する因子負荷量

	Fac.1	Fac.2	Fac.3	Fac.4	Fac.5	Fac.6	Fac.7
1.1	-0.096	0.235	0.054	0.195	0.173	-0.067	0.759
1.2	0.072	0.911	-0.005	-0.024	0.112	-0.001	-0.148
1.3	-0.008	0.958	0.030	0.017	0.006	-0.025	-0.084
1.4	-0.039	0.793	-0.085	0.239	0.273	0.118	0.030
2.1	-0.064	0.547	0.292	-0.350	-0.050	0.116	0.353
2.2	0.213	0.460	0.541	-0.055	-0.404	-0.198	0.066
2.3	0.035	0.071	0.592	0.072	0.387	0.282	-0.001
2.4	0.082	0.040	0.414	-0.391	0.190	0.058	0.518
2.5	-0.149	-0.095	0.808	0.137	-0.113	0.087	0.018
3.1	0.078	0.011	0.142	0.144	0.027	0.504	0.348
3.2	0.102	-0.097	0.435	0.135	0.070	0.591	-0.052
4.1	-0.092	0.032	0.214	0.766	-0.048	0.247	-0.120
4.2	0.076	0.014	-0.025	0.447	0.000	0.343	0.447
4.3	0.370	0.001	-0.035	0.686	-0.057	-0.248	0.171
4.4	0.294	0.071	-0.008	0.162	-0.031	0.636	-0.019
4.5	0.611	0.000	-0.179	0.015	0.183	0.480	-0.077
4.6	0.863	-0.024	-0.057	0.010	0.009	0.161	-0.010
5.1	-0.041	0.358	-0.170	-0.176	-0.263	0.554	0.382
5.2	-0.022	0.705	-0.115	0.014	-0.222	0.002	0.349
5.3	-0.084	0.043	0.098	-0.128	-0.074	0.822	-0.058
5.4	0.060	0.073	-0.015	-0.094	0.864	-0.115	0.104
5.5	0.602	-0.162	-0.018	0.180	-0.007	0.188	0.423
5.6	0.898	0.110	0.110	-0.004	-0.042	-0.220	-0.128

表3: 37プロジェクトに対する因子スコア

PR	Fac.1	Fac.2	Fac.3	Fac.4	Fac.5	Fac.6	Fac.7
1	1.619	0.927	1.080	0.500	-2.112	-2.109	2.035
2	0.961	1.070	-1.085	-1.590	0.568	1.518	1.099
3	-0.989	0.105	2.205	0.258	1.475	0.327	-0.725
4	0.671	1.587	0.992	-1.018	-0.735	0.125	-1.248
5	1.180	0.728	1.643	0.760	1.130	1.031	0.016
6	-0.969	1.855	-0.404	1.781	-0.095	-0.062	0.976
7	-1.110	-1.196	1.316	-0.190	0.947	1.174	1.962
8	0.557	1.054	1.555	0.719	0.304	1.727	0.549
9	2.124	-0.280	-0.490	-0.617	1.315	-1.166	1.264
10	-0.090	1.104	1.275	1.048	0.610	0.027	-1.156
35	0.141	-0.955	0.649	-1.642	0.802	-0.924	0.697
36	-0.381	-0.939	-0.581	-0.028	-0.911	-0.888	-0.041
37	-0.194	-0.805	1.421	-1.902	-1.900	-0.032	0.692

表2に示す因子負荷量を元に、それぞれの因子の意味を考える。ここでは因子2について説明する。表2からアンケート項目の1.3, 1.2, 1.4が因子2に強く関わっている。表1から、1.3は「要件の理解力不足」、1.2は「要件の引出し力不足」、1.4は「要件に対する顧客側、実現側相互の合意不足」である。これらから、因子2は「要求仕様段階での危険性」を表すと判断し、名前付けをした。同様に、残りの因子についても名前付けをし、次の7つの因子を得た。

- 因子1: 進捗管理に関する危険性
- 因子2: 要求仕様段階での危険性
- 因子3: 見積り段階での危険性
- 因子4: 工程計画に関する危険性
- 因子5: 要件や仕様の変更の管理に関する危険性
- 因子6: プロジェクトの体制に関する危険性
- 因子7: プロジェクトの目標設定に関する危険性

### 4.3 Step 3 の結果 (モデルへの適用)

ステップワイズ法に基づき、上で求めた因子1~7の因子スコア(それぞれ  $x_1 \sim x_7$  で表す)を用いてロジスティック回帰式を求めた。その結果、因子5以外の全ての要因が有意であると判断された。

$$P(y|x_1, x_2, x_3, x_4, x_6, x_7) = \frac{e^{2.16 - 1.93x_1 - 1.96x_2 - 1.77x_3 - 0.97x_4 - 1.44x_6 - 1.78x_7}}{1 + e^{2.16 - 1.93x_1 - 1.96x_2 - 1.77x_3 - 0.97x_4 - 1.44x_6 - 1.78x_7}}$$

これは、抽出された6つの因子 ( $x_1, x_2, x_3, x_4, x_6, x_7$ ) がいずれも「プロジェクトの混乱」を決定する重要な要因であることを示している。

[注1] 文献 [2] のモデルの場合には2つの因子だけが利用されていたが、これが6つの因子にまで拡張できた。

## 5. 評価実験

表3の因子スコアと4.3のロジスティック回帰式からプロジェクトが混乱する条件付き確率を求める。その値が0.5以上であれば混乱する、そうでなければ混乱しないと予測する。予測の結果を表4に示す。この結果から37プロジェクトの中で34個について正しい予測ができることを確認した。[2]での同様の実験結果に対して約8%の精度向上が確認された。

表4: 37プロジェクトに対する予測結果

Actual	Predicted	
	Confused	Not Confused
Confused	10	1
Not Confused	2	24

## 6. むすび

本報告では、因子分析に基づくアンケート調査表の集約と、それを用いた混乱予測法についての検討を行った。その結果、プロジェクトの混乱に直接影響を及ぼすと考えられる7つの因子を明らかにできた。また、これらの因子を用いたモデル式を利用すれば、プロジェクトの混乱予測が非常に高い精度で行えることを確認した。

## 参考文献

- [1] B.W.Boehm, "Industrial software metrics top 10 list", IEEE Software, vol.4, no.5, pp.84-85, 1987.
- [2] O. Mizuno, T. Kikuno, Y. Takagi, and K. Sakamoto, "Characterization of risky projects based on project managers' evaluation," Proc. of 22nd International Conference on Software Engineering, pp.387-395, 2000.
- [3] 芝祐順, 因子分析法, 東京大学出版会, 1986.
- [4] E. Yourdon, *Death March: The Complete Software Developer's Guide to Surviving 'Mission Impossible' Projects*, Prentice Hall Computer Books, 1997.