

タスクの並列度を考慮した ソフトウェア開発工程の遅延対策

橋浦研究室

1155346 西舘 香那

1. はじめに

大規模なソフトウェアの開発は労働力を結集するためにプロジェクトを組んで行われるのが一般的である。ソフトウェア開発プロジェクトには開発計画(= 開発のための作業スケジュールや各作業への要員割当などに関する計画)というものがある。現在、ソフトウェア開発の現場では、大規模で複雑なソフトウェアを短納期かつ低コスト、高品質で開発する要請が多い。加えて、開発されるソフトウェアの大規模化・複雑化傾向はますます進み、ソフトウェア開発の高生産性・高信頼性が求められている[1][2]。

2. 研究目的

ソフトウェア開発プロジェクトには、原因は様々であるが開発工程遅延が度々発生する。その度にプロジェクトマネージャーはソフトウェア開発計画、工程遅延回復案、対策案の立案等に多くの時間を費やしている。本研究は工程遅延回復案の立案の手間を省くために、後続作業に対して人員を追加することによる工程遅延回復案を自動立案する新たな手法を提案する。

3. 提案手法

ブルックス[3]は、ソフトウェア開発のように仕事が連続していて分担できない場合、人を増やすという工程遅延対策はスケジュールリング上効果が薄いことを指摘している。これを緩和するために、本研究ではソフトウェア開発工程の並列度を考慮し、人員を追加することで工程遅延回復する工程を算出し、回復案の自動立案を行う。

3. 1 工程の並列度

ソフトウェア開発工程の中には、並列化に向いている工程と不向きな工程が存在する。例えば、要件定義では仕事の分割は不向きであるが、プログラム工程やテスト工程はモジ

ュールごとなど複数人に仕事の分割をすることができ、並列化に向いているといえる。

3. 2 アムダールの法則 [4]

アムダールの法則とは、ある計算機システムとその対象とする計算について、その計算機の並列度を上げた場合に、全体として期待できる全体の性能向上の程度を数式として表現したものである。並列化可能部分の割合を P 、並列化不可能部分を $(1-P)$ 、プロセッサ数を N とすると、速度向上比 $S(N)$ は式(1)で与えられる。

$$S(N) = \frac{N}{P + N(1-P)} \quad (1)$$

本研究では、この法則を前述の並列度を考慮した工程遅延の回復日数の計算に利用する。プロセッサ数を人員 M (人) とすると、式(2)より所要時間の減少率 $\left(\frac{1}{S(N)}\right)$ を導くことができる。

$$\frac{1}{S(N)} = \frac{P + N(1-P)}{N} \quad (2)$$

ここで、ある工程の所要日数を d としたとき、回復日数 R は式(3)で求められる。

$$R = d - \left(d \times \frac{1}{S(N)}\right) \quad (3)$$

例えば初期計画において作業者が 1 人、並列度が 50%、所要日数が 8 日の工程に人員を 1 人追加する場合、 P は 0.5、 N は 2、 d は 8 となる。これを(3)に代入すると、 $\frac{1}{S(N)} = 0.75$ 、 $R = 2$ と求まる。よってプロジェクト計画は現状の終了予定日から 2 日回復することが分かる。

4. 実験

作成したソフトウェア開発計画において工程遅延が発生したと仮定する。そのとき、当初の計画の期日に満足できるような工程遅延回復案を提案することができるかどうかを検証する。

今回の実験では、小規模プロジェクト(3~6ヶ月)[5]、中規模プロジェクト(1~2年)[5]の場合に分けて検証を行う。工程の遅延について

は各プロジェクトに対して人工的にプロジェクトの 1~2%の遅延を発生させ、その場合の回復案をシステムによって立案する。今回実験に用いた小規模プロジェクトの計画全体の所要期間の平均日数は 41 日、1~2 日の遅延は計画全体の 2~4%、中規模プロジェクトの計画全体の所要期間の平均日数は 215 日、4~5 日の遅延は計画全体の 1~2%の遅延である。

本研究で取り扱うプロジェクトは全て以下の 3 つの条件を満たしたものとした。

- (1). すべてのタスクには前後関係が存在する
- (2). 余裕日やファストトラッキングを考慮しない
- (3). 追加する人員は 1 名、かつ対策する工程は 1 工程である

4. 1 リスク

対策案を実施するタイミングにより、考慮しなければならないリスクは変化する。例えば、開発計画の前半部分(30%)、中盤部分(50%)、終盤部分(75%)で対策を実施する場合を考える。まず、30%時点では、その後、さらに遅延が発生した場合でも、再度対策案の再計画をすることができる。50%時点では、30%時点よりも対策可能な工程が少なくなるため、再度対策することが困難になる。75%時点の場合は、さらにその傾向が顕著になる。従って、対策のタイミングが遅くなるほど失敗する可能性が低い方法をプロジェクトマネージャーは選択しなければならない。

工程遅延回復案において、所要日数と並列度の割合は異なるが、回復日数が同じ場合の工程が存在する。このような場合において、本研究では、プロジェクトマネージャーに対し、①現在日に近い工程、もしくは②並列度が高い工程を優先する 2 つのリスク選択を提示する。前者は、現在日に遠い工程に人員を追加した場合、対策に失敗したとき納期に間に合わなくなる可能性を、後者は、並列度が小さい工程は仕事の分割が困難であるため人員を追加しても失敗する可能性が高くなること考慮したものである。

4. 2 評価

当初終了日からの遅延日数を L 、回復日数

を R とすると、当初終了日との誤差 D は以下の式(4)で求められる。

$$D = L - R \quad (4)$$

$\begin{cases} D > 0 & \text{工程遅延回復効果が薄い} \\ D \leq 0 & \text{工程遅延回復効果がある} \end{cases}$

従って本研究では、 $D \leq 0$ のようなプロジェクト計画を立案することが目的となる。

4. 3 回復効果の考察

実験に用いた開発計画事例の一例を付録 A に示す。この計画は要求分析からテスト工程までの 12 の工程からなる計画である。リスク①、②を考慮した回復日数を R_r 、 R_b 、当初終了日との誤差を D_r 、 D_b とする。

30%の時点で対策した結果を表 1 に示す。表 1 より 30%の時点で対策を実施すると、3 日未満の遅延($L < 3$)までは $D \leq 0$ となっており、遅延日数 2 日までは回復案の立案が可能であることが分かる。現在日に近い工程を優先したい場合は工程#3、並列度が高い工程を優先したい場合は工程#5 に追加することを提案する。

50%の時点では、以前として遅延日数 2 日までは回復案の提案が可能である。75%の時点では、対策できる工程が#6 に限定され、再度の対策は不可能であるので、30%や 50%の時点と比較した場合、期日に間に合わなくなる可能性が高いことが分かる。

一方で、計画全体の 75%の時点で対策を実施しても、回復効果が薄い立案しか行えない事例も明らかになった。その事例を表 2 に示す。表 2 より、30%、50%の時点では、2 日未満の遅延($L < 2$)までは $D \leq 0$ となっており、遅延日数 1 日までは回復するが、75%の時点で対策を実施しても当初の計画の納期に間に合う回復案を提案することは不可能である。

今回用意した小規模プロジェクトの事例におけるプロジェクト別最大回復量(プロジェクト全体の日数に対する回復量の割合)を表 3 に示す。1 つのプロジェクトには、複数対策案ができるので、その回復日数の最大値を R_{max} とする。表 3 より 75%の時点で対策をしても当初の期日に間に合わないプロジェクトは D、E、G の 3 事例(43%)、また 75%の時点で対策すると回復案を提案するが、最後の工程に対

策を実施することを提案する事例が A, B の 2 事例(29%)存在した. これより工程遅延が発生したとき, 計画全体の 75%の時点で対策を実施することは当初の期日に間に合わない可能性が高いことが明らかになった.

表 1 回復効果(小規模プロジェクト)

| L (日) | 工程 | Dr (日) | 工程 | Dh (日) |
|-------|----|--------|----|--------|
| 1 | #3 | -1 | #5 | -1 |
| 2 | #3 | 0 | #5 | 0 |
| 3 | #3 | 1 | #5 | 1 |
| 4 | #3 | 2 | #5 | 2 |
| 5 | #3 | 3 | #5 | 3 |

表 2 回復効果(中盤までに対策必須)

| L (日) | 工程 | Dr (日) | 工程 | Dh (日) |
|---------|-----|--------|-----|--------|
| < 30% > | | | | |
| 1 | #10 | -0.25 | #10 | -0.25 |
| 2 | #10 | -0.75 | #10 | -0.75 |
| | | | | |
| 1 | #10 | -0.25 | #10 | -0.25 |
| 2 | #10 | -0.75 | #10 | -0.75 |
| | | | | |
| 1 | #11 | 0.25 | #11 | 0.25 |
| 2 | #11 | 1.25 | #11 | 1.25 |

表 3 小規模プロジェクトの対策別回復効果

| PJ | R _{max} | | |
|----|------------------|-----|-----|
| | 30% | 50% | 75% |
| A | 5% | 4% | 4% |
| B | 2% | 2% | 2% |
| C | 3% | 3% | 3% |
| D | 3% | 3% | 0% |
| E | 1% | 1% | 0% |
| F | 3% | 3% | 3% |
| G | 5% | 5% | 0% |
| 平均 | 3% | 3% | 2% |

5. まとめ

ソフトウェア開発計画において工程遅延が発生したとき, 後続作業に対して人員を追加することによる回復案の有効性を確認することができた.

今後の課題として, 実際のプロジェクトにおいて 1 人で作業した場合の所要した日数と, 2 人で作業した場合の所要した日数の差を取ることにより, 実際の開発工程の並列化可能部分の割合を逆算することが可能である. 実際の開発工程の並列度がわかれば, より明確な工程遅延回復案の立案が可能になる.

参考文献

- [1] 木下 大輔, 林 雄一郎, 八重樫 理人, 上之 蘭 和弘, 橋浦 弘明, 古宮 誠一, “制約に基づくソフトウェア開発計画自動立案システム-システムを実装するための戦略と評価-,” 情処研報, Vol. 2004, No. 87(2004-SE-145), pp. 1-8, Aug. 2004.
- [2] 高須賀 公紀, 嶋村 彰吾, 内川 裕貴, 木下 大輔, 林 雄一郎, 古宮 誠一, “余裕日も考慮に入れたソフトウェア開発の自動立案,” 情報処理論文誌, Vol. 48, No. 8, pp. 2713-2723, Aug. 2007.
- [3] フレデリック・P・ブルックス Jr., “人月の神話 新装版,” 株式会社 ピアソン エデュケーション, pp. 11-23, Oct. 2003.
- [4] Gene M. Amdahl, “Validity of the single processor approach to achieving large scale computing capabilities,” AFIPS spring joint computer conference, pp. 1-4, 1967.
- [5] エドワード・ヨードン, “デスマーチ,” 第 2 版, 日経 BP 社, pp. ix-5, Aug. 2006.

付録 A

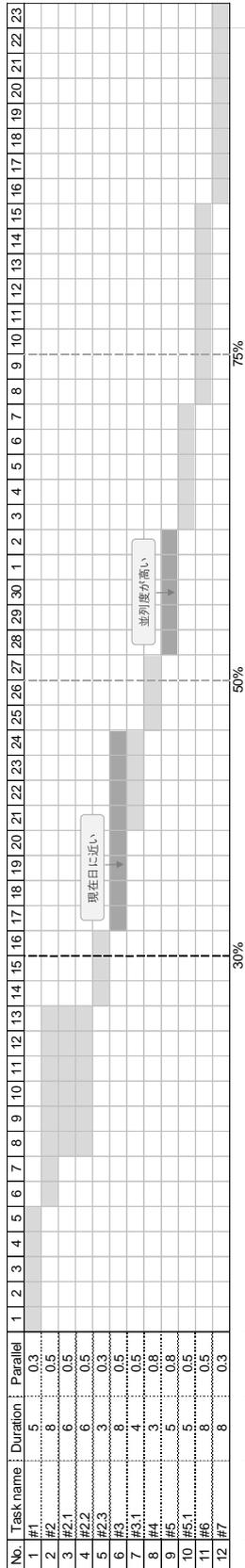


図 1 要件定義からテスト工程までの小規模プロジェクト