R-3

タスクの粒度を制限するプロジェクト管理支援ツール

橋浦研究室　　1155414　細井 愛斗

1. はじめに

現在，多くの大学のシステム開発実習授業で PBL(Project-Based Learning，以下 PBL)を導入している[1][2][3]．しかし，PBL のプロジェクトは学生の開発計画の問題によって失敗することがある[4]．

2. PBL における学生の問題

学生は演習の一環として取り組む PBL より前に，プロジェクトベースの開発の経験を積んでいる場合が少ない．そのため，学生は作業を細かく分けることが困難であり，個々のタスクの粒度が粗くなってしまう．これを確認するために，本学の工学部情報工学科の 4 年生 7 名のタスクを対象に調査を行った．その結果を図 1 に示す．これを見ると，粒度の粗いタスクは大きい予実誤差が発生しやすいことが分かる．

図 1. 見積工数と誤差時間の関係

3. 研究目的

本研究の目的は，前述の問題を軽減することで，大きな予実誤差が発生しやすい開発計画の作成を防ぐことである．加えて，終わりが見えない作業の対応を学生が行いやすくすることで，期日内に終了できるようにする．

本研究における終わりが見えない作業とは，見積工数を超えて作業していても進捗が芳しくないタスクのことを指す．このようなタスクは，学生が解決できない問題があり，この問題の解決に時間を要している可能性がある．そのため，早めに教授者からフィードバックを得ることで遅延を軽減する．

4. 提案手法

本研究では，学生が作成するタスクの予定工数に上限を設けることで，粒度が粗いタスクを作成できないようにする．また，遅延が発生している作業に対処するために，教授者から作業のフィードバックを得るように促す．なお，進捗が芳しくなく，かつ期日が近いタスクほど遅延状況が改善される可能性が低くなるため，このような作業を優先してフィードバックを得るよう促す．

5. 実現方法

前述の手法を実現するためにはタスク管理ツールが必要である．そのため，本研究では Redmine[5]を利用する．さらに，専用のプラグインを追加するとともに，Redmine のカスタマイズを行う．

Redmine とは，オープンソースで開発されているプロジェクト管理ツールである．Redmine では作業をチケットとして定義し，チケットには予定工数や開始日などの予定や，行った作業の内容とその経過時間などの実績を登録することができる．チケットに関連する機能は標準でもさまざま備わっており，プラグインも多く開発されている．

本研究の提案手法を実現するためには，Redmine に以下の機能が必要となる．

① 学生の作成するチケットの予定工数を 3 時間までに制限する機能
② 遅延が発生したチケットの情報を学生が教授者に通知する機能

①の機能は，図 2 のようなエラー文が表示され，チケットが登録されなくなる機能である．②の機能は，チケットに設定されている担当者が更新された際に通知されるようにし，作業が遅延した際にチケットの担当者を教授者に変更するように学生に指示することで利用させる．図 3 に通知内容の例を示す．この通知内容のうち，作成者，チケット名，チケットの説明を見ることで，教授者は誰のどの

作業に対してフィードバックを行う必要があるかが分かる．遅延チケットの確認は利用者が行うため，遅延チケットは一覧表示し，分かりやすくしている(図 4)．一覧には，期日が間近で，進捗率が低いチケットが上部に表示されるようにしている．これは 4 節で述べた通りである．学生は，この一覧の中から作業時間が予定工数を超過しているチケットを探し，見つかれば，そのチケットの担当者を更新することで教授者に通知を行う．

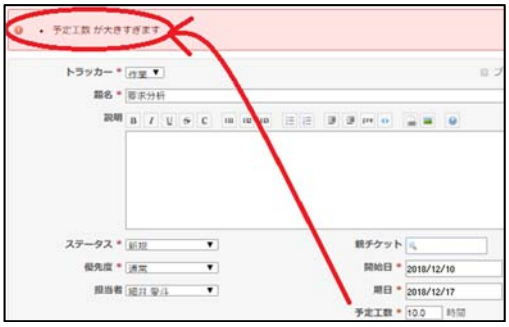

図 2. 上限を超える予定工数を設定したときのエラー

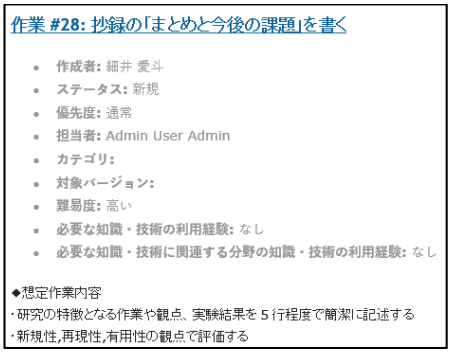

図 3. 通知されるチケットの内容例

図 4. 遅延チケット一覧表の例

6. 実験

本研究の提案手法によって，作業の遅延・誤差がどれほど軽減され，チケットを期日内に終了できるようになるのかを明らかにするために実験を行った．被験者は，本学の工学部情報工学科の 3 年生 9 名である．被験者には，システム開発実習や研究室における作業を対象に作業計画を立て，Redmine 上に登録してもらう．加えて，被験者には作業の性質をチケットごとに設定してもらう．考慮する作業の性質を表 1 に示す．作業を行う際には時間を計測してもらい，行った作業の内容と作業時間を Redmine 上に登録してもらう．実験期間は 2 週間とし，1 週目には前述の機能が実装されていない Redmine を，2 週目には機能が実装された Redmine を利用し，その差を比較することとした．

表 1. 作業の性質

番号	性質	程度		
①	難易度	高い	普通	低い
②	必要な知識・技術の利用経験の有無	あり	なし	
③	必要な知識・技術に関連する分野の技術・技術の利用経験の有無	あり	なし	

7. 実験結果と考察

実験では，1 週目に 8 個，2 週目に 24 個の合計 32 個のチケットが作成された．性質ごとにまとめた作成チケット数と期日超過チケット数を表 2 に示す．また，実験期間別の期日超過チケット数と期日超過率を表 3 に示す．

表 2. 性質別チケット

項番	性質			チケット数		期日超過チケット数	
	①	②	③	1週目	2週目	1週目	2週目
1	低い	あり	あり	4	2	0	0
2	低い	あり	なし	–	–	–	–
3	低い	なし	あり	–	–	–	–
4	低い	なし	なし	–	–	–	–
5	普通	あり	あり	1	10	0	0
6	普通	あり	なし	–	2	–	0
7	普通	なし	あり	–	1	–	0
8	普通	なし	なし	2	4	2	0
9	高い	あり	あり	1	5	2	0
10	高い	あり	なし	–	–	–	–
11	高い	なし	あり	–	–	–	–
12	高い	なし	なし	–	–	–	–

表 3. 実験期間別の期日超過チケット

		実験期間	
		1週目	2週目
期日	無超過	4	0
	超過	4	24
期日超過率		50%	0%

まず，期日内に作業を終了できるようになったかどうかを評価する．表 3 より，実験 1 週目の期日超過率は 50%で，2 週目は 0%であったが，この期日超過率の差に有意差は確認できなかった．ここで，実験 1 週目に期日を超過したチケットに着目する．このチケットは表 4 で示す通りである．実験 1 週目には，表 4 のチケット 2 と 4 のように，実験 1 週目の平均予定工数 3.47 と比較して大きい予定

工数が設定されているものがあった．このチケットを作成した被験者の実験 2 週目のチケットを表 5 に示す．表 5 より，この被験者は実験 2 週目で作業の開始日と期日までの日数が短くなっており，すべてのチケットが期日までに終了していた．しかし，大半のチケットは予定工数を超過していた．表 5 のチケット 3 のような遅延が続くと，期日超過のリスクがある．つまり，予定工数を制限し，作業の粒度を細かくするだけでは取り除けない期日超過に繋がるリスクがあると考えられる．

表 4. 実験 1 週目に期日を超過したチケット

チケット	性質			開始日	期日	終了日	予定工数
	①	②	③				
1	高い	あり	あり	9月20日	10月18日	12月6日	3.00
2	高い	あり	あり	10月18日	12月13日	未終了	15.00
3	普通	なし	なし	12月3日	12月5日	12月6日	2.00
4	普通	なし	なし	11月1日	12月13日	未終了	15.00

表 5. ある被験者の 2 週目のチケット

チケット	性質			開始日	期日	終了日	予定工数	実工数
	①	②	③					
1	高い	あり	あり	12月6日	12月13日	12月13日	2.00	3.00
2	普通	あり	なし	12月6日	12月13日	12月13日	1.00	2.00
3	高い	あり	あり	12月6日	12月13日	12月13日	2.00	20.00
4	普通	あり	あり	12月6日	12月13日	12月13日	3.00	1.50
5	高い	あり	あり	12月6日	12月13日	12月13日	3.00	5.00

次に，作業の予実遅延・誤差が軽減されたかを評価する．図 5 より，本研究の手法を適用した週の方が遅延割合が低かったが，有意差は確認できなかった．誤差割合においてはほぼ差が見られなかった(図 6)．遅延割合は下記の(1)，誤差割合は(2)の式で計算した．

$$\text{遅延割合} = \frac{\text{実工数}-\text{予定工数}}{\text{予定工数}} \quad (1)$$

$$\text{誤差割合} = \left|\frac{\text{実工数}-\text{予定工数}}{\text{予定工数}}\right| \quad (2)$$

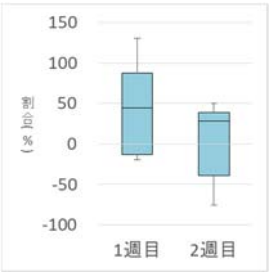

図 5. 作業の遅延割合

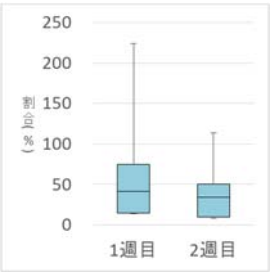

図 6. 作業の誤差割合

今回の実験では，遅延チケットの情報が 1 件も通知されなかった．考えられる要因は，作業の粒度が細かくなったことで，作業が遅延しても，想定している残りの作業は少なく，フィードバックを得る必要はないと被験者が判断してしまったためだと考えられる．その結果，表 6 に示す通り，遅延割合 50%以上の遅延に繋がってしまっている．このことから，作業の粒度が細かすぎると，予定工数の倍近く遅延していてもフィードバックを得ようと思わなくなってしまうことがあるため，予定工数には下限を設ける必要があると考える．

表 6. 実験 2 週目の遅延チケット

チケット	性質			予定工数	実工数	遅延割合
	①	②	③			
1	高い	あり	あり	2.00	3.00	50%
2	高い	あり	あり	2.00	20.00	900%
3	高い	あり	あり	3.00	5.00	67%
4	高い	あり	あり	0.50	0.98	96%
5	普通	あり	あり	0.50	1.07	114%
6	普通	あり	なし	1.00	2.00	100%

8. まとめと今後の課題

本研究では，PBL における学生の開発計画の予定と実績が大きく乖離してしまう問題に着目し，問題の軽減を実現するための機能を実装した Redmine を用いた実験を行った．その結果，予定工数に上限を設けることで，粒度が粗すぎることが原因の遅延が取り除かれ，作業が期日を超過してしまうリスクを軽減できる傾向があることが分かった．しかし，作業の粒度が細かすぎるとフィードバックの機会は減少し，逆に大きな作業遅延に繋がることが判明した．よって，予定工数には下限も必要であることが明らかになった．

今後は，作業の性質ごとに評価を行えるようにするために，実際の PBL で実験を行うなどで被験者を増やし，実験期間を長くすることで，実験の規模を大きくする必要がある．また，遅延作業の通知は被験者の判断が介される方法ではなく，ある時点で自動的に行われる方法で行う必要がある．

参考文献

[1] 茨城大学, “2017 年度 シラバス ソフトウェア開発演習Ⅱ,” 2017/2, (参照 2018/3/22).
[2] 筑波大学, “平成 30 年度 情報科学類（共通） 科目一覧 ソフトウェア工学,” 2018/2, (参照 2018/8/3).
[3] 宇都宮大学, “2017 年度 シラバス システム設計演習 I (PBL),” 2017/2 , (参照 2018/3/22).
[4] 齋藤 尊, 新美 礼彦, 伊藤 恵, “過去の情報を用いた PBL 向け工数見積り手法の提案,” コンピュータソフトウェア, Vol.35, No.1, pp.117-123, 2018/2.
[5] Jean-Philippe Lang, “Redmine,” <https://www.redmine.org/>, 2018/6/10, (accessed 2018/12/20).