
プログラミング実習におけるグループ開発プロセスの分析

An Analysis of Group Development Processes in a Programming Class

玉田 春昭* 井垣 宏† 引地 一将‡ 門田 暁人§ 松本 健一¶

Summary. In this paper, we quantitatively analyzed the relation between software products and communication among developers based on the experimental study of group development. In the software development, each group consists of 4 to 5 students who learned the object oriented design and Java language. As a result, all the unsuccessful groups showed lower communication time than the average. Also, there were no unsuccessful group whose communication time was very high.

1 はじめに

多くのソフトウェア開発現場において、開発作業はグループワークであり、多人数が協調して作業を進めることで、納期までにソフトウェアを完成・出荷させることが可能となる。このような多人数での開発は、一人での開発と比べると、作業の切り分けと割り当て、作業者間のコミュニケーション、進捗管理など、グループワーク特有の作業が必要となる。そのため、学生が実社会に出る前に、プログラミング言語や設計方法について学ぶのみならず、グループワークのプロセスやその重要性について学ぶことが重要となる [1, 2]。

我々は、ソフトウェア工学教育の一環として、学部3年生を対象にグループ開発の体験学習を行った。学生は実際の開発経験を通じて、グループワークのプロセスやその重要性について学習する。ただし、グループ開発を体験するだけで、学生がその重要性を習得できるとは限らない。また、教員にとっても、グループワークの評価を行うことは容易でない。なぜなら、成果物の出来栄が良いからといって、作業者間の協調がうまく行われていたとは限らないためである。そこで、グループ開発のプロセスを評価し、良かった点や反省点を学生に理解させるために、プロセスについて何らかの定量的なデータを収集・分析し、グループワークの重要性を示す客観的な分析結果を学生にフィードバックすることが望ましい。

我々は、ある大学の講義において行ったグループ開発の終了時に、開発プロセスに関して学生にアンケート調査を行うことで、プロセスの詳細を知るための定量的なデータを23個のグループから得た。そして、完成したソフトウェアに対する評価と、開発プロセスとの関係を分析した。本稿では、グループ開発実験の詳細とその分析結果について報告する。

2 実験

2.1 実験計画

プログラミング実習の講義中にグループ開発を行わせ、実習終了後にアンケート調査を行った。開発させたアプリケーションは Web アプリケーションであること

*Haruaki Tamada, 奈良先端科学技術大学院大学 情報科学研究科

†Hiroshi Igaki, 奈良先端科学技術大学院大学 情報科学研究科

‡Kazumasa Hikichi, 奈良先端科学技術大学院大学 情報科学研究科

§Akito Monden, 奈良先端科学技術大学院大学 情報科学研究科

¶Ken-ichi Matsumoto, 奈良先端科学技術大学院大学 情報科学研究科

表 1 講義内容

第 1 回	第 2 回	第 3 回	第 4 回	第 5 回	第 6 回	第 7 回	第 8 回
ガイダンス	UML	Web クライアント	Web サーバ	データベース	JDBC	Servlet	JSP

という制限以外は特に設けず，各グループごとに要求仕様から立てさせた．

講義は全 14 回行い，その前半 8 回を用いて UML や Servlet, JSP, データベースの講義を実習を交えて行い，残り 6 回をグループ開発実施に充てた．講義で行った内容は表 1 の通りである．

そして，グループ開発の実習時に，その時間に作成すべき中間成果物とその意味について 30 分ほど講義を行った．その後，学生にその中間成果物を作成させ，次週までに提出させた．作成させた中間成果物は企画書，ユースケース図，クラス図，ページ遷移図，シーケンス図，インストールガイド，テスト計画書の 7 つである．

そして，最終成果物として，アプリケーションのソースコード（画面の Web ページ含む）を提出させた．また，グループごとに作成したアプリケーションを発表する時間を設け，その発表資料も成果物として提出させた．

被験者はある大学の情報系学部の 3 年生とし，その数は 225 名であった．全員が Java 言語基本の講義を受講している．加えて本実験と同時期にオブジェクト指向設計に関する講義とデータベースの講義を受講している．

2.2 成果物の評価

成果物は教員 3 名，TA 3 名が完成度，有用性，独創性，技術性という 4 つの観点で評価し，その合計を総合点とした．それぞれの配点は 2 点である．

それぞれ評価する点は以下の通りである．

完成度 要求仕様に書かれている機能が実装されているかどうか

有用性 完成したアプリケーションが有用なものかどうか

独創性 独創的なアプリケーションかどうか

技術性 新しい技術やライブラリを使用しているかどうか

2.3 アンケート

実習終了時に学生全員にアンケートに答えてもらった．アンケートの質問項目は以下の通りである．

- 担当した作業（複数回答可）
- ミーティング回数
- ミーティング 1 回あたりにかけた時間
- ミーティングで主に議論した相手（複数回答可）．
- 開発を行った時間

まず，グループ内での役割分担を調べるため，中間成果物と最終成果物のうち担当したものを回答してもらった．

また，グループ開発であるので，ミーティングによる意思疎通が大切である．そのため，ミーティングを計何回行ったのか，また，1 回あたりどのくらいの時間を費やしたのか，そして，ミーティングで主に議論した相手を尋ねた．この議論した相手を回答者の議論相手とする．最後に，一人で開発を行った時間を調べ，グループのメンバの開発時間の違いを調べた．

このアンケート結果から，グループメンバそれぞれをノード，あるメンバのノードからそのメンバの議論相手に向かって矢印を引いて有向グラフを作成する．これをコミュニケーショングラフと呼ぶ．また，グループのコミュニケーション密度として，コミュニケーショングラフの辺密度を用いる．辺密度はグラフ G が持ちえる辺全体に対する実際の辺の割合であり， $\frac{G \text{ の辺の数}}{\text{完全グラフの辺の数}}$ で表される [3]．ここで，コミュニケーショングラフは有向グラフであるので，辺が両向きであった場合は 2 本の辺と数える．

表 2 アンケート集計結果

	グループ	人数	コミュニケーション密度	ミーティング		成果物の評価				
				回数	総時間	完成度	有用性	獨創性	技術性	総合点
上位 25%	A	4	1.00	10	10	12	12	10	10	44
	B	4	0.33	1	2	12	11	11	9	43
	C	5	0.88	4	5	12	10	10	7	39
	D	4	1.00	6	36	11	11	8	8	38
	E	4	1.00	20	60	11	10	7	8	36
	F	5	0.15	6	27	9	9	8	9	35
下位 25%	R	5	0.95	3	4.5	8	8	4	5	25
	S	5	0.65	3	6	8	6	7	2	23
	T	5	0.55	3	6	9	5	5	4	23
	U	5	0.60	5	10	5	7	5	3	20
	V	5	0.50	4	8	5	4	5	3	17
	W	4	1.00	4	6	5	4	3	2	14

表 3 各メンバの開発時間

	グループ	メンバ					合計	平均	分散
		a	b	c	d	e			
上位 25%	A	3	10	20	3		36	9.00	48.50
	B	10	10	4.5	70		94.5	23.63	721.92
	C	24	10	10	3	10	57	11.40	49.44
	D	10	1	2	1		14	3.50	14.25
	E	80	200	15	10		305	76.25	6108.25
	F	45	50	5	10	5	115	23.00	406.00
下位 25%	R	1.5	1.5	1.5	1.5	1	7	1.40	5.44
	S	3	1	2	3	3	12	2.40	0.64
	T	2	15	5	3	16	41	8.20	36.56
	U	6	10	8	6	10	40	8.00	3.20
	V	2	1.5	12	3	1.5	20	4.00	15.46
	W	5	2	2	12		21	5.25	16.69

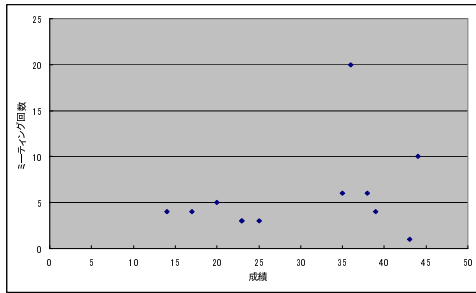


図 1 成績とミーティング回数の関係

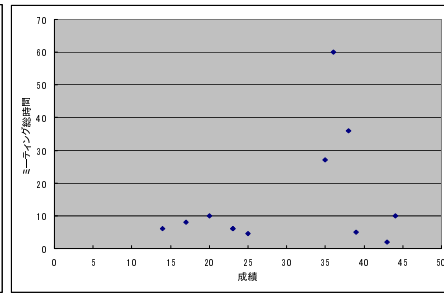


図 2 成績とミーティング総時間の関係

3 実験結果と分析

3.1 実験結果の概略

アンケートの有効回答数は 188 で、メンバ全員が有効回答者であるグループは 23 グループであった。このグループのうち、総合点の上位 25% の 6 グループと下位 25% の 6 グループの集計結果を表 2 に、また、各グループのメンバそれぞれの開発時間とそのグループごとの合計、平均、分散を表 3 に示す。また、各人の担当した作業を表 4 (上位 25%) と表 5 (下位 25%) に示す。

表 2 から、成績の上位 25% に含まれるグループのミーティング回数やミーティングの時間にばらつきがあることがわかる。また、コミュニケーション密度も 2 極化していることがわかる。反対に、成績の下位 25% に含まれるグループはコミュニケーション密度が 0.5 ~ 0.65 と中程度のグループが多い (4/6)。また、コミュニケーション密度と成果物の評価との間の相関は見られなかった。

図 1 と図 2 に成果物の総合点とミーティング回数、ミーティング総時間の関係を散布図で表す。図 1 の横軸は成績で、縦軸はミーティング回数である。同じように、図 2 の横軸は成績で、縦軸はミーティングの総時間を表す。図 1, 2 から成績の低いグループではミーティングの回数が少なく、ミーティングの時間も少ないことがわかる。

成果物の評価点数の良いグループのいくつかは作業時間の突出したメンバがいることがわかった。また、そのようなメンバが不在のグループでも議論をよく行い、役割分担を明確にすることで良い成果物を作ることが十分可能であることがわかった。

対して、成果物の評価点の悪いグループは、コミュニケーションの回数、総時間ともに平均以下であることがわかった。また、作業時間の突出したメンバもあらず、これらのグループでは、役割分担が明確になっていないケースも確認された。

3.2 上位 25%に含まれるグループに関する分析

上位 25% に含まれるグループのうち、コミュニケーション密度が 1 ではない B, C, F グループのコミュニケーショングラフを図 3, 4, 5 に示す。

総合点がトップであった A グループは作業分担ができていた。ソースコード作成は A_c のみであるが、テストはメンバ全員で行っており、ソースコードを作成していないメンバもテストを行っている。これは他のグループに見られない特徴であり、完成度が満点となっているのは、これが一つの要因であると考えられる。

B グループはミーティングの回数が 1 回、総時間が 2 時間と、全グループと比較しても圧倒的にミーティングの回数、総時間が少なくなっている。また、図 3 より、 B_d はグループの中心人物であることがわかる。また、表 4 より、多くの作業が B_d により行われている。このことから、 B_d が他の 3 人を管理し、また、自らも開発するという B_d の高い能力が総合点が高い要因であると考えられる。ただし、各作業者の勤務時間が定められた現実のソフトウェア開発では、一人の間人だけが他の作業者の何倍もの時間を費やして開発を行うことは現実的ではない。

C グループは C_a の開発時間が高くなっている。また、表 4 の担当作業を見ると C_a と C_e がソースコードとテストを行っている。図 4 の関係図から各メンバの重要度を PageRank アルゴリズム [4] により算出した。その結果は $C_a = 0.216, C_b = 0.173, C_c = 0.243, C_d = 0.108, C_e = 0.259$ であった。このことから C グループでのミーティングでの主要なメンバは C_a, C_c, C_e であることがわかる。また、表 4 の担当作業と重要度を比較すると、上位工程である企画書を担当しているのは重要度が高い二人であることがわかる。また、 C_e はそのまま C_a と協力してソースコードの作成、テスト作業に入り、 C_b が C_c の補佐的な役割として作業に参加しているように見える。

D グループは D_a のみがソースコード作成を行い、それ以外は各成果物に対して 2 人以上のメンバが担当となっている。また、1 回当たりのミーティングの時間が 6 時間と全グループ中一番長く、総ミーティング時間も長い。技術性がそれほど高くなく、開発時間の合計も低いことから、プログラミングが得意な人物がいないと考えられる。しかし、全員で協力した結果、上位 25% に入ったのだと考えられる。

E グループは E_b が中心となり開発を行っているように見える。ソースコード作成に E_a と E_c が E_b の補佐役にまわり、 E_d は資料作成に回っているように見える。 E_b の作業時間が 200 時間と飛びぬけて大きく、 E_b が多大な時間を費やしたことで良い成果物が出来たと考えられる。ただし、前述のように、現実のソフトウェア開発の勤務体系では、このような開発形態は現実的ではない。

F グループでは、ミーティングで議論を行っていない F_a, F_b がソースコード作成を行っており、 F_c, F_d, F_e と一切関わりを持っていない。また、表 4 から、 F_d と F_e が企画書を作成してから F_a と F_b にソースコードの作成を依頼しているように見え、発注側と下請けのような関係になっている。

3.3 下位 25%に含まれるグループに関する分析

R グループと S グループは開発時間の合計がそれぞれ 7 時間、12 時間となっており、下位 25% に属するグループの中でも極端に少なく、また開発時間の分散も低い。加えてミーティングの総時間も多くない。そのため、開発のために時間を割いてはあらず、良い成果物にならなかったと考えられる。

An Analysis of Group Development Processes in a Programming Class

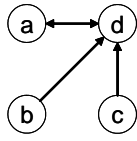


図3 Bグループ

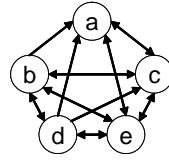


図4 Cグループ

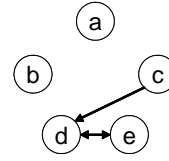


図5 Fグループ

表4 グループメンバの担当作業 (上位 25%)

グループ名	A				B				C				
メンバ	a	b	c	d	a	b	c	d	a	b	c	d	e
企画書	✓			✓	✓	✓	✓	✓			✓		✓
ユースケース	✓			✓	✓	✓	✓	✓		✓	✓		
ページ遷移図	✓		✓		✓	✓					✓		
クラス図			✓					✓		✓	✓		
シーケンス図			✓		✓	✓	✓			✓	✓		
Web ページ				✓	✓			✓		✓	✓		
ソースコード			✓		✓			✓	✓				✓
テスト	✓	✓	✓	✓				✓	✓				✓
マニュアル				✓		✓							✓
発表資料		✓			✓	✓		✓					✓

グループ名	D				E				F					
メンバ	a	b	c	d	a	b	c	d	a	b	c	d	e	
企画書		✓	✓	✓	✓	✓	✓	✓					✓	✓
ユースケース	✓	✓	✓	✓		✓				✓				
ページ遷移図			✓	✓				✓	✓					
クラス図	✓			✓		✓			✓					✓
シーケンス図	✓	✓	✓	✓		✓								
Web ページ	✓			✓		✓				✓			✓	✓
ソースコード	✓				✓	✓	✓	✓	✓	✓				✓
テスト	✓	✓			✓	✓	✓	✓	✓	✓				✓
マニュアル				✓		✓		✓						✓
発表資料		✓	✓	✓				✓					✓	✓

表5 グループメンバの担当作業 (下位 25%)

グループ名	R					S					T				
メンバ	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e
企画書	✓	✓			✓			✓							✓
ユースケース			✓	✓				✓					✓		
ページ遷移図			✓	✓		✓					✓		✓		
クラス図				✓									✓		✓
シーケンス図				✓			✓						✓		✓
Web ページ				✓	✓					✓		✓			✓
ソースコード				✓				✓	✓	✓	✓	✓			✓
テスト				✓				✓		✓	✓	✓			✓
マニュアル	✓					✓						✓	✓		✓
発表資料	✓	✓	✓		✓				✓	✓					✓

グループ名	U					V					W			
メンバ	a	b	c	d	e	a	b	c	d	e	a	b	c	d
企画書	✓			✓		✓		✓						✓
ユースケース			✓	✓		✓	✓						✓	✓
ページ遷移図			✓						✓	✓			✓	
クラス図			✓						✓				✓	
シーケンス図		✓									✓			
Web ページ				✓				✓		✓				
ソースコード			✓	✓	✓		✓		✓	✓				
テスト				✓		✓		✓	✓	✓	✓	✓		
マニュアル	✓			✓							✓	✓		✓
発表資料				✓		✓					✓			✓

W グループはソースコード作成担当者がおらず、アプリケーションの主要部分の担当者を特に定めなかったと考えられる。担当作業を見ると、 W_d が非常に多くの作業を担当しており、開発時間も W_d が他のメンバに比べて高くなっている。しかし、上位 25%に含まれるグループの場合と異なり、プログラミングに関する能力はそれほど高くなかったと考えられる。

4 考察

以上は、プログラミング実習でのグループ開発に関する議論であった。この実験結果と同じような現象が実業務で起こった場合、どのような解釈になるかを考察する。

まず、ミーティングの回数が少ないグループが上位下位グループともに存在した。特に上位 25%に含まれる B グループは下位 25% に含まれるグループと比べても極端にミーティングの回数と総時間共に少ない。今回のグループ開発は、小規模ソフトウェアの開発であったため、ミーティング総時間が少ない場合においても、一人の学生が大部分の作業を行うことで良い成果物ができることがあったが、より大規模なソフトウェア開発ではうまくいかないことを学生に指導する必要がある。つまり、実業務でミーティング総時間がこのように少ない場合、仕様変更への対応などが疎かになるため、近い将来、そのプロジェクトは失敗に終わる可能性が高いため、ミーティングにより相互理解を図ることが重要であると指導する必要がある。

次に、各メンバの開発時間の分散に関して考察する。上位 25% に含まれるグループは開発時間の分散が非常に高くなっている。例えば、B, E, F が挙げられる。E グループについては開発時間の最長と最短のメンバで開発時間が 20 倍違っている。実業務における勤務体系では、このような作業時間のばらつきは非現実的であることを学生へ指導することが重要となる。仮に、実業務において作業時間にばらつきがでた場合は、今回の結果とは逆に、開発時間の長い作業者の能力が疑われることになるだろう。そのため、開発時間の突出は悪であり、作業時間が均等になるように作業を分担させるように指導しなければならない。

グループ開発という観点から見ると開発時間の分散が高い、すなわち、少数の人間の作業時間が突出していることは決して容易に受け入れるべきものではなく、実業務においても非現実的である。グループのメンバ全員が均等に開発し、コミュニケーション密度も高く保つことが理想のグループ開発である。そのため、プログラミング実習という枠内で、グループ開発の経験のない学生たちを、如何に実業務に近い形のグループ開発に近づけていくことを考える必要がある。

5 まとめ

ある大学の講義において行ったグループ開発について、開発プロセスと成果物の評価の関係を分析した。その結果、ミーティングの回数が多いグループ、もしくはミーティングの総時間が多いグループに成果物の評価が低いグループは存在しないことがわかった。また、成果物の評価点が良いグループでは技術的リーダーの存在があることが多いことがわかった。

今後の課題として、成果物に対する静的な評価値を加えた考察が挙げられる。また、アンケートの回答内容の妥当性の向上のため、アンケート実施の事前周知が必要である。

参考文献

- [1] 松浦佐江子. グループワークによるソフトウェア開発に向けたオブジェクト指向教育実践の試み. 情報処理学会 研究報告 — コンピュータと教育, 第 2003 巻, pp. 85–92, 12 2003.
- [2] 松浦佐江子, 相場亮. グループワークによるソフトウェア工学教育の試み. 情報処理学会 研究報告 — コンピュータと教育, 第 2003 巻, pp. 1–8, 2 2003.
- [3] R. ディーステル. グラフ理論. シュプリンガー・フェアラーク東京, Oct 2000. ISBN4-431-70876-6.
- [4] Larry Page, Sergey Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.