

経済産業省委託調査

大学等におけるIT教育実態調査報告書  
(情報系学科卒業生の視点)

平成 16 年 3 月

IT教育実態調査プロジェクト  
(株式会社テクノフェイス)



# 目次

<b>1. はじめに</b>	<b>3</b>
<b>2. 調査にあたって</b>	<b>5</b>
<b>3. 調査方法</b>	<b>7</b>
3.1. アンケート対象者	7
3.2. スケジュール	7
3.3. アンケート文	7
<b>4. アンケート結果</b>	<b>8</b>
<b>5. 分析と考察</b>	<b>139</b>
5.1. アンケート回答者	139
5.2. 講義・演習の方法と効果	140
5.2.1. 学校別の学習科目の特徴	140
5.2.2. 職業別の役立っている科目	142
5.2.3. 科目別特徴について	148
5.2.4. 理解と役立ち	150
5.2.5. 勉強方法と理解	151
5.2.6. 勉強方法と効果	152
5.2.7. 印象に残っている科目	153
5.3. 講義・演習の方法と満足度	154
5.3.1. 授業方法と満足度	154

5.3.2.	授業科目と過不足.....	155
5.3.3.	PBL.....	156
5.3.4.	インターンシップ.....	157
5.3.5.	PBL、インターンシップでの習熟 .....	157
5.3.6.	授業以外でのシステム開発経験 .....	158
5.3.7.	卒論・修論・博士論文 .....	159
5.3.8.	学会・論文の経験と効果.....	160
<b>5.4.</b>	<b>企業内教育.....</b>	<b>162</b>
5.4.1.	教育内容と習熟度.....	162
<b>5.5.</b>	<b>業務を通しての習熟度.....</b>	<b>163</b>
<b>5.6.</b>	<b>現在の業務.....</b>	<b>164</b>
<b>5.7.</b>	<b>研修または自分での勉強 .....</b>	<b>165</b>
<b>5.8.</b>	<b>資格 .....</b>	<b>166</b>
<b>5.9.</b>	<b>大学教育への要望.....</b>	<b>167</b>
<b>5.10.</b>	<b>所属する企業.....</b>	<b>168</b>
<b>5.11.</b>	<b>所属する学会 .....</b>	<b>168</b>
	<b>「添付資料：アンケート文」.....</b>	<b>170</b>

# 1. はじめに

---

韓国では、ICU (Information & Communications University) が脚光を浴びているようだ。大場等の報告<sup>1</sup>に以下の実情が述べられている。

- ・ ソフトウェア技術者育成のために国の支援で設立された国立でない若い大学。
- ・ '98年、大学院 114 人でスタート。'02年、学部生 105 人でスタート。  
学部生が約 100 人、大学院生が約 100 人位 (半分は海外留学生：主に東南アジア)。
- ・ 選抜は、国のトップ 1% (SAT) から応募の面談と優秀な高校推薦者の面談で決める。
- ・ 全員スカラシップあり。大学院も TA、RA を多く採用し、殆ど勉学に専念可能。
- ・ CMU (Carnegie Mellon 大学) と提携し、1 セメスターをカーネギーメロンの Software Engineering Course で過し、ダブルメジャーが可能。18 人/年位が CMU に行く。教官も 1 人帯同。
- ・ 教授 60 人、その他の教官 31 人、スタッフ 39 人、合計 130 人と充実している。  
英語で講義すると+100 万円/年、国立でなくても教官たちの給与を高くできる。
- ・ 高校の秀才 (gifted student) 10 人を大学で特別教育している。

これも、現時点の韓国国内 IT 関連総生産額の 10% 強であるといわれる海外向け輸出額をインド並みの 80% にしたいとの国の戦略の一部だという。

ところで、我が国でも主要大学で大学院レベルでの新しい情報関連教育が稼動し出した。我が国の国是である「科学技術立国」との連関性においても情報関連大学院への期待は大きなものがある。IT バブルがはじけたといえ、それはあくまでもビジネスの世界の瞬間的な出来ごとであって、産業界のみならず、われわれの生活に関連するすべての分野において、今後ますますその重要性を増しつつあるであろう IT の持つ基本的特性とは無縁であろう。

IT は人間の脳機能の人工的拡大・増大機能の実現手法ともいえる。これの安易な比喻として自動車を取り上げてみたい。人間の身体的特性のひとつである骨格・筋力に起因した固体移動能力を拡大・増大する機能が自動車の持つ基本的機能であり、それは今やわれわれの生活空間にその存在は不可欠のものである。この自動車そのものを海外から輸入してこれを活用することのみに徹するのか、それとも我が国が自動車産業王国になるのかが問題である。自動車は多種多様な技術の集大成でもあり、もしも自動車王国化を指向した場合、優れた技術者集団の育成と確保はその大前提となる。IT 分野でも韓国は後者を意識している様子がうかがえよう。急がば回れの感があるが、何事においてもことをなすには人

---

<sup>1</sup> 大場,他; 韓国大学訪問報告, 2004年3月.

材育成が肝要であろう。これまでの IT 人材育成を担ってきた、我が国の高等専門教育機関である高専、大学の学部、大学院での IT 関連の教育はそろそろ 40 年の歴史を刻む。

その間、多種多様な現場経験をもつ多くの卒業生を輩出してきた。今後の IT 人材育成のあり方を模索するためにも、かつて当事者であった卒業生の実体験に基づいた意見は貴重である。以上の意識のもとここにアンケート調査を行った結果を報告する。

## 2. 調査にあたって

---

種々の分野の基盤技術としての IT はますますその重要性を増しつつありますが、これを担うべき IT 人材に関して、産側が必要とする IT 関連技術者と、学側が育成してきた IT 人材の間には超え難い溝がありますのは周知のことと推察いたします。

国是として、“IT を中心とした科学技術創造立国により、世界への貢献を目指す” わが国の IT に関する現状は、憂慮すべき状況に陥っているのではないのでしょうか。

このような状況は一朝一夕にして生ずるものではなく、これまでの教育機関における IT 教育のあり方、企業の IT 人材育成戦略にその遠因を求めるべきでありましょう。

現状を打破すべき試みとして、北海道大学では平成 15 年 4 月より、産学連携による IT トップガン人材育成事業がスタートしています。このような試みが数多く行われれば、結果として現状打破につながるのではないかと期待されます。

しかし、一般的にはいまや産側と学側の IT 教育に関する認識は、お互いに共通の言語を持ち得ないと疑わざるを得ない悲劇的レベルにあるのではないのでしょうか。

国是を成就させるために“IT 立国”という車の両輪をなす産学の緊密な連携は必須かと思えます。両輪を連結する車軸として“産学連携共同研究”も大事ですが、学から産にゆるく結合できるような“連続性のある IT 人材育成教育”は、その重要性をもっと認識されても良いのではないのでしょうか。

さらに言えば現時点での産側の要求に応えるのみではなく、世界をリード可能な IT 人材をいかにして育成していくか、などの議論があっても良いのではないのでしょうか。IT 技術による世界への貢献を目指して。

ところで“連続性”はどうなっているのでしょうか。残念ながら IT 教育というキーワードで産学間の連続性を議論するに足る十分なデータを見つけることは困難です。

具体的データなしに問題解決のための議論が展開できるはずもありません。データがあれば、対処策を模索・立案・実行することが可能になるはずです。

このような貴重なデータはどこにあるのでしょうか。

学で専門教育を受け、産で種々の実体験をお持ちの第一線で活躍されている皆様、IT 関連技術者がお持ちのはずです。ただしそのデータは具体的な形にはなっていないと思います。

本調査は、皆様へのアンケートを通して皆様の暗黙データを統計量として具現化し、これらに基づいて、卒業生から見た学側におけるこれまでの問題点を明らかにし、もって今後の展望のためのひとつの基礎データを提供することを目的としたものであります。

具体的には、(1)大学での教育が個人の能力に如何に定着し、実社会で役立っているか、役立とうとしているか、(2)企業内教育(OJT、研修会、自己研鑽、知的交流)で能力向上させ、実業務で如何に役立っているか、についてアンケート方式により調査を行おうとするものです。

調査にあたっては、経済産業省のご支援の下、調査プロジェクトを立ち上げ、皆様のご理解とご協力によってアンケート実施・分析を進めていくものであります。何卒、主旨にご賛同いただき皆様の貴重なご意見を賜りますようお願い申し上げます。

プロジェクト名称：IT教育実態調査プロジェクト

プロジェクトリーダー：嘉数侑昇（教授）

住所：〒060-8628 札幌市北区北13条西8丁目

北海道大学大学院 工学研究科 システム情報工学専攻

複雑系工学講座 自律系工学分野

## 3. 調査方法

---

### 3.1. アンケート対象者

アンケートは、主に平成7年度以降に情報系の高専・大学学部・大学院を卒業・修了したIT関連企業の社員（現職の専門は問わず）を対象とした。

### 3.2. スケジュール

2月4日：アンケート用紙一式の配布

※アンケート回答用紙の返信は、個人情報に関する配慮のため、「返信封筒」を使用し、回答者ご本人から直接事務局へ送付していただいた。

3月3日：回答回収の完了

3月8日：集計の完了

分析作業、報告書作成

3月15日：報告

### 3.3. アンケート文

配布したアンケートの内容は、「添付資料：アンケート文」を参照のこと。

## 4. アンケート結果

---

今回は北海道大学寄附講座企業の情報に関連する社会人を中心にアンケートに参加していただいた。アンケートは書類郵送またはメール方式を用いて行い、短期間ではあったが、646名（回答率11.6%：646回収／5,585配送）の多くの方からの回答を得ることができた。アンケート文は16頁に渡り、膨大な質問項目にも関わらず、11.6%にもおよぶ回答率を得たことは、多くの方がIT関連の教育に高い関心と期待を寄せていることがうかがえる。

まず、アンケート回答の基本データを整理した図表を次頁以降にまとめる。分析と考察については5章で述べる。

## 「アンケート回答の基本データ」

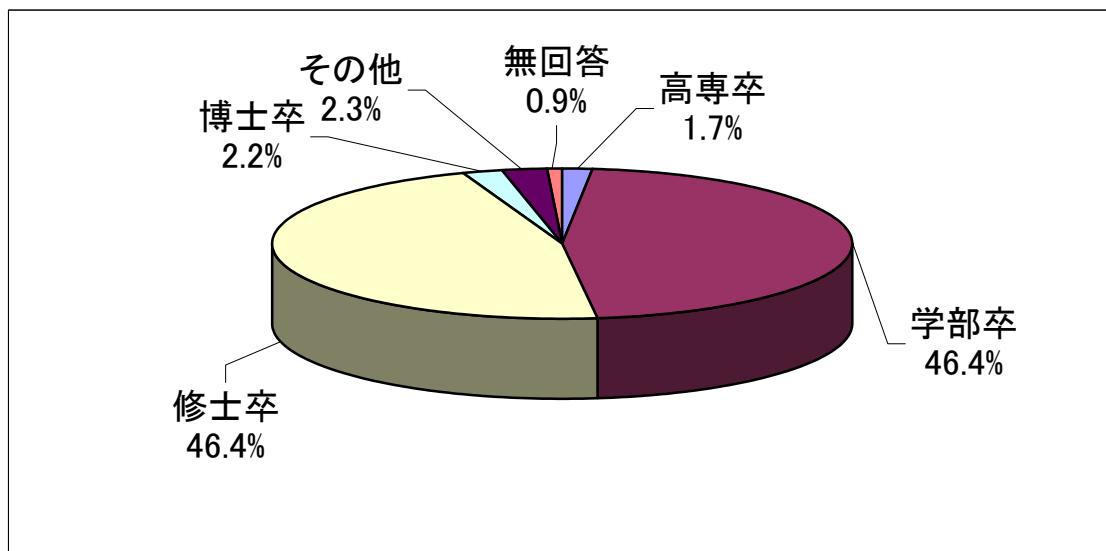
1.	記入者について	10
2.	講義・演習で学んだことについて	17
3.	教育方法について	54
4.	企業内教育（情報工学関連の教育）について	97
5.	業務としてのシステム／ソフトウェア開発を通しての習熟度について	102
6.	現在の業務について	119
7.	現在、研修または自分で勉強していることについて	123
8.	資格（取得済み・目指している資格）	130
9.	要望	133
10.	所属する企業について	135
11.	所属する学会などについて	137

## 1. 記入者について

- 1－a－1. 最終学歴
- 1－a－2. 卒業後経過年数
- 1－a－3. 卒業学科・研究科・専攻
- 1－a－4. 現在の業務内容
- 1－a－5. 最終学歴
  
- 1－b－1. 大学と最終学歴
- 1－b－2. 業務と卒業学科・研究科・専攻
- 1－b－3. 現在の業務内容と卒業後経過年数

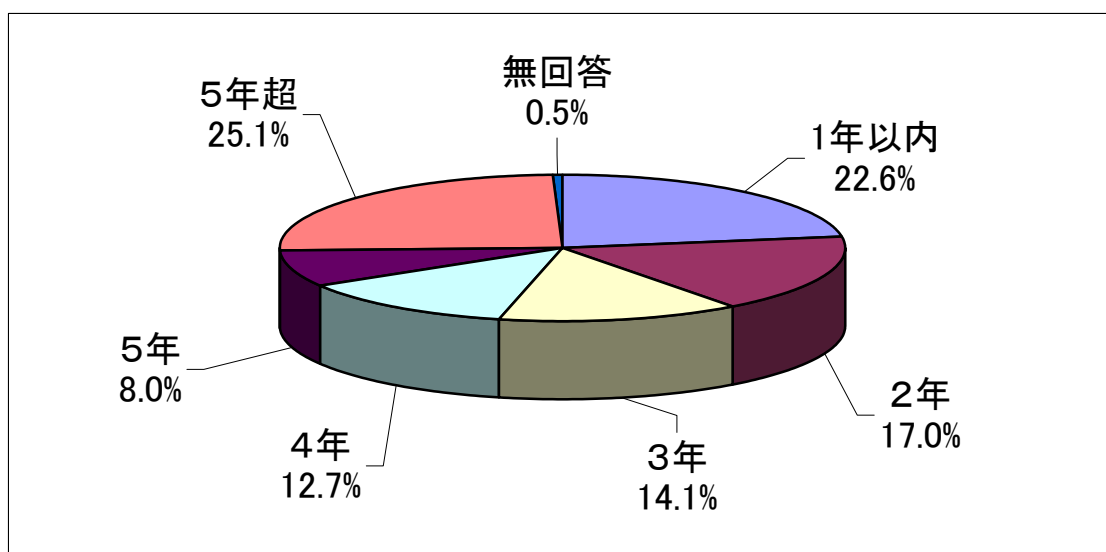
### 1-a-1. 最終学歴

最終学歴	人数	比率
高専卒	11	1.7%
学部卒	300	46.4%
修士卒	300	46.4%
博士卒	14	2.2%
その他	15	2.3%
無回答	6	0.9%
合計	646	



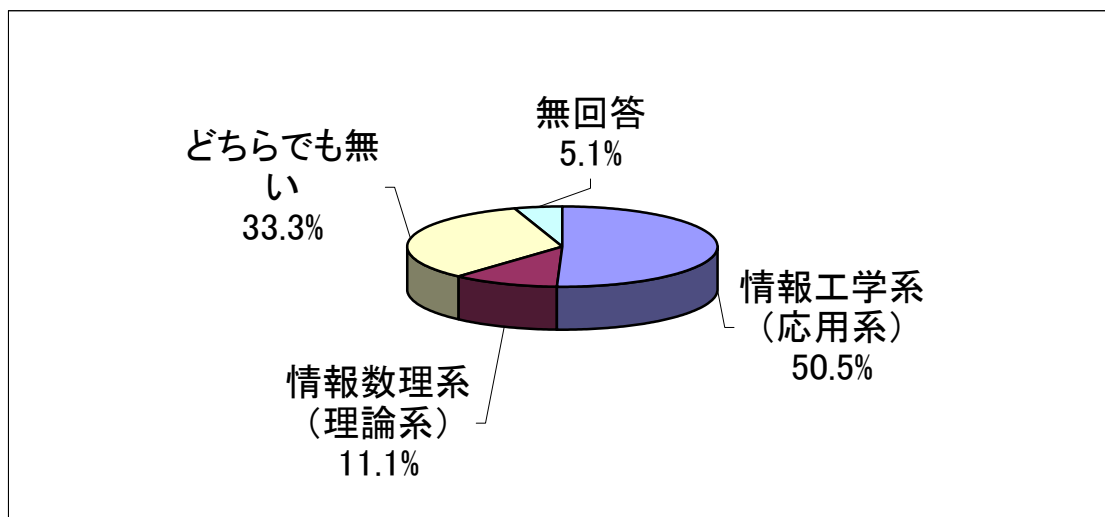
### 1-a-2. 卒業後経過年数

卒業後経過年数	人数	比率
1年以内	146	22.6%
2年	110	17.0%
3年	91	14.1%
4年	82	12.7%
5年	52	8.0%
5年超	162	25.1%
無回答	3	0.5%
合計	646	



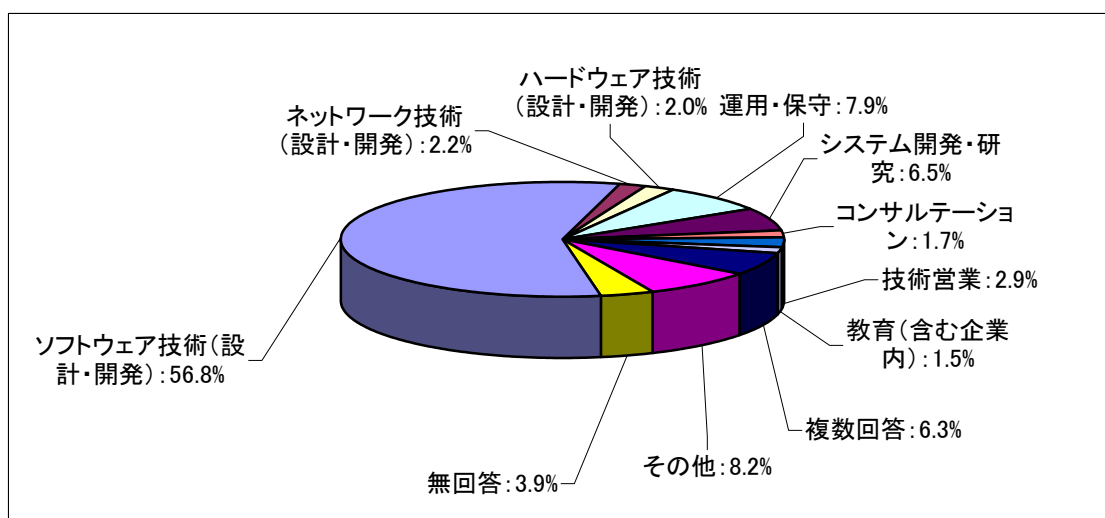
### 1-a-3. 卒業学科・研究科・専攻

卒業学科・研究科・専攻	件数	比率
情報工学系(応用系)	326	50.5%
情報数理系(理論系)	72	11.1%
どちらでも無い	215	33.3%
無回答	33	5.1%
合計	646	



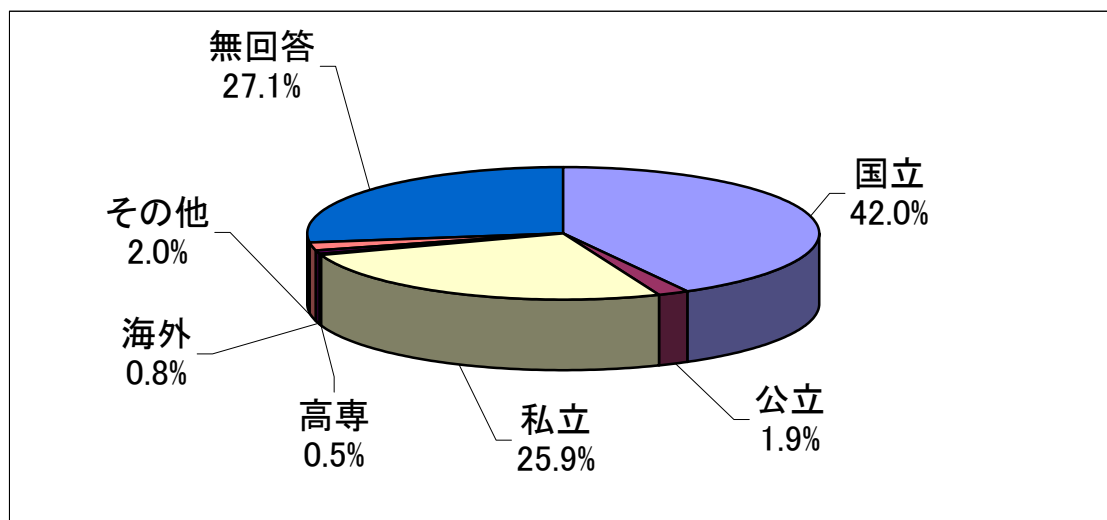
### 1-a-4. 現在の業務内容

業務	件数	比率
ソフトウェア技術(設計・開発)	367	56.8%
ネットワーク技術(設計・開発)	14	2.2%
ハードウェア技術(設計・開発)	13	2.0%
運用・保守	51	7.9%
システム開発・研究	42	6.5%
コンサルテーション	11	1.7%
技術営業	19	2.9%
教育(含む企業内)	10	1.5%
複数回答	41	6.3%
その他	53	8.2%
無回答	25	3.9%
合計	646	



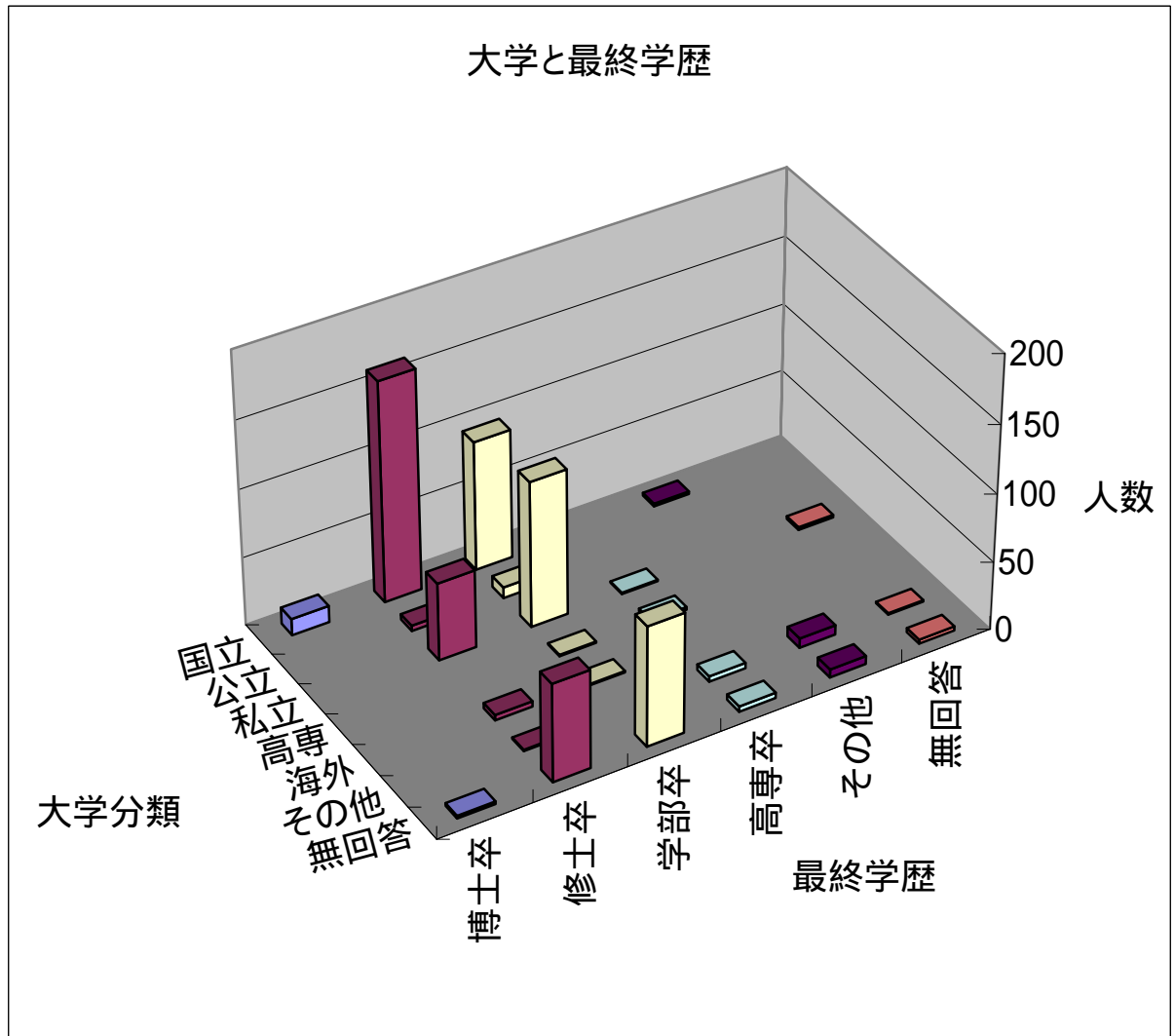
### 1-a-5. 最終学歴

大学分類	合計	比率
国立	271	42.0%
公立	12	1.9%
私立	167	25.9%
高専	3	0.5%
海外	5	0.8%
その他	13	2.0%
無回答	175	27.1%
合計	646	



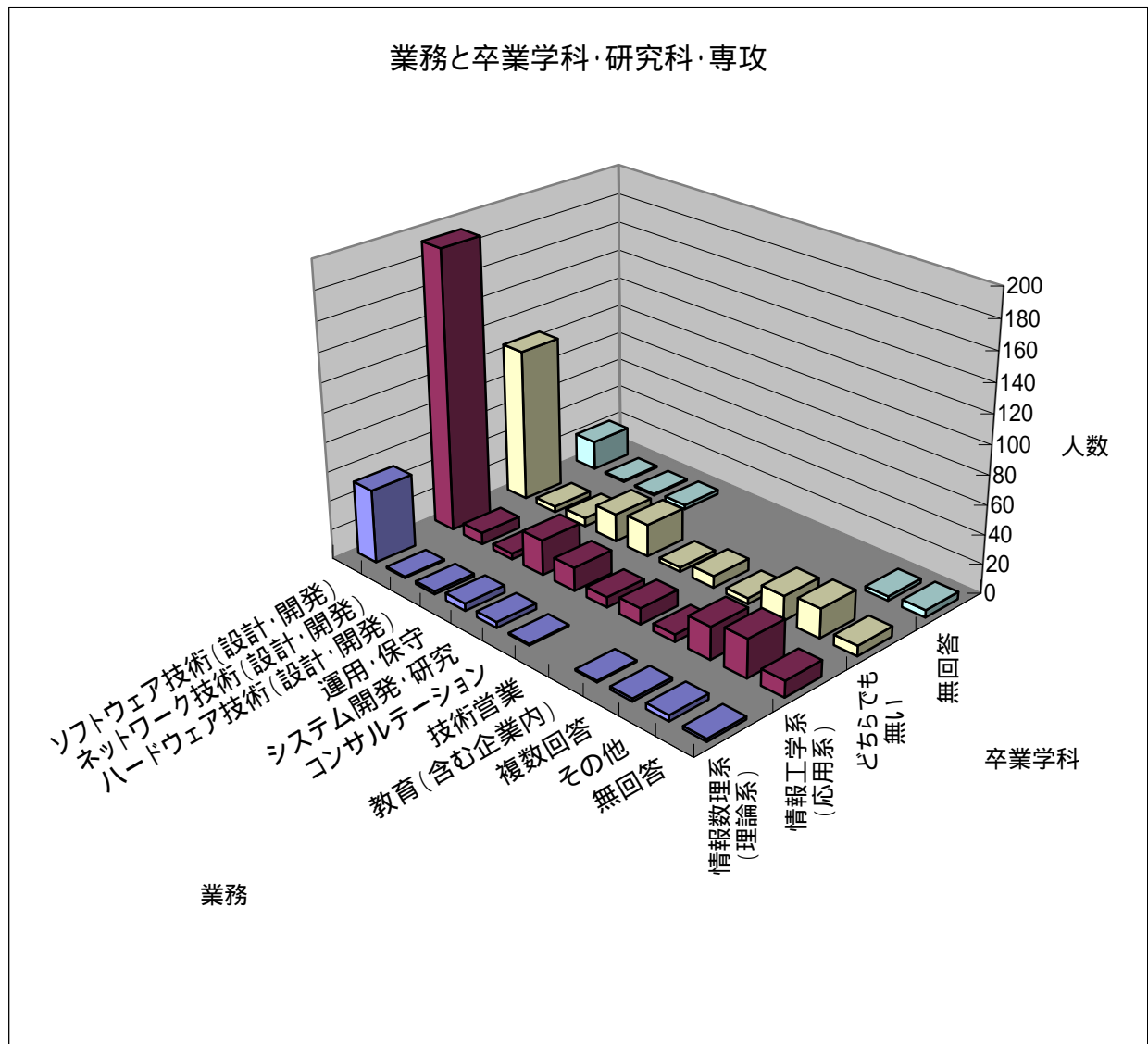
# 1 - b - 1 . 大学と最終学歴

		最終学歴					合計
		博士卒	修士卒	学部卒	高専卒	その他	
大学分類	国立	12	162	95		2	271
	公立		4	8			12
	私立		57	107	1		167
	高専			1	2		3
	海外		4	1			5
	その他		1		4	7	13
	無回答	2	72	88	4	6	175
合計		14	300	300	11	15	646



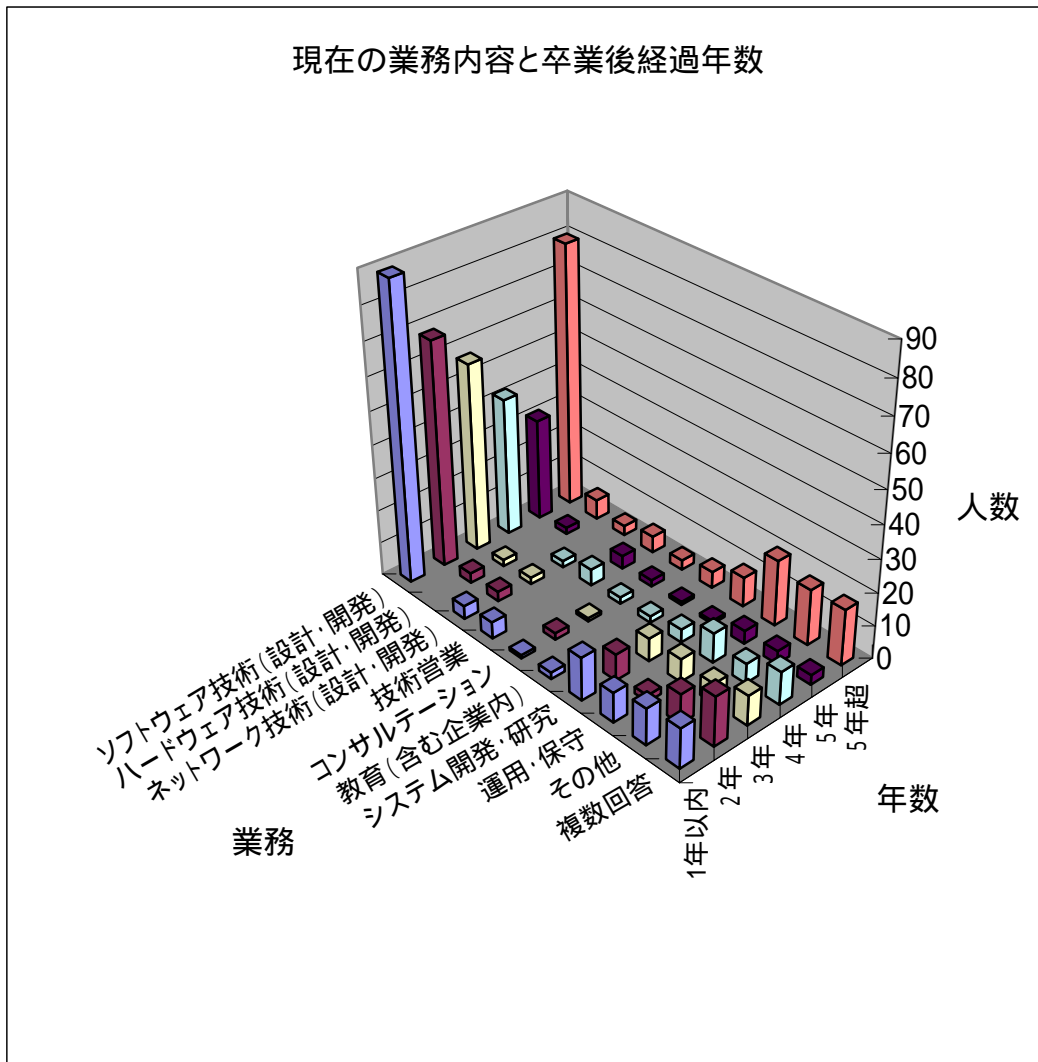
# 1 - b - 2 . 業務と卒業学科・研究科・専攻

		卒業学科・研究科・専攻				合計
		情報数理系 (理論系)	情報工学系 (応用系)	どちらでも 無い	無回答	
業務	ソフトウェア技術(設計・開発)	50	192	105	20	367
	ネットワーク技術(設計・開発)	1	8	4	1	14
	ハードウェア技術(設計・開発)	2	4	6	1	13
	運用・保守	5	24	19	3	51
	システム開発・研究	4	16	22		42
	コンサルテーション	1	7	3		11
	技術営業		11	8		19
	教育(含む企業内)	1	5	4		10
	複数回答	2	22	17		41
	その他	4	26	20	3	53
	無回答	2	11	7	5	25
	合計	72	326	215	33	646



### 1 - b - 3 . 現在の業務内容と卒業後経過年数

	卒業後経過年数						合計
	1年以内	2年	3年	4年	5年	5年超	
ソフトウェア技術(設計・開発)	89	68	57	42	31	80	367
ハードウェア技術(設計・開発)		3	2		2	6	13
ネットワーク技術(設計・開発)	4	3	2	2		3	14
技術営業	5			5	4	5	19
コンサルテーション	1	2	1	2	2	3	11
教育(含む企業内)	2			2	1	5	10
システム開発・研究	13	8	7	4	1	9	42
運用・保守	9	2	7	9	4	20	51
その他	11	9	6	6	4	17	53
複数回答	12	15	9	10	3	17	46
合計	146	110	91	82	52	165	646



## 2. 講義・演習で学んだことについて

### 2-1. 講義・演習の理解度、効果について

2-1-a-1. 勉強方法（全体）

2-1-a-2. 理解度（全体）

2-1-a-3. 効果（全体）

2-1-b-1. 勉強方法と理解：プログラミング入門  
～

2-1-b-9. 勉強方法と理解：ソフトウェア工学

2-1-c-1. 勉強方法と効果：プログラミング入門  
～

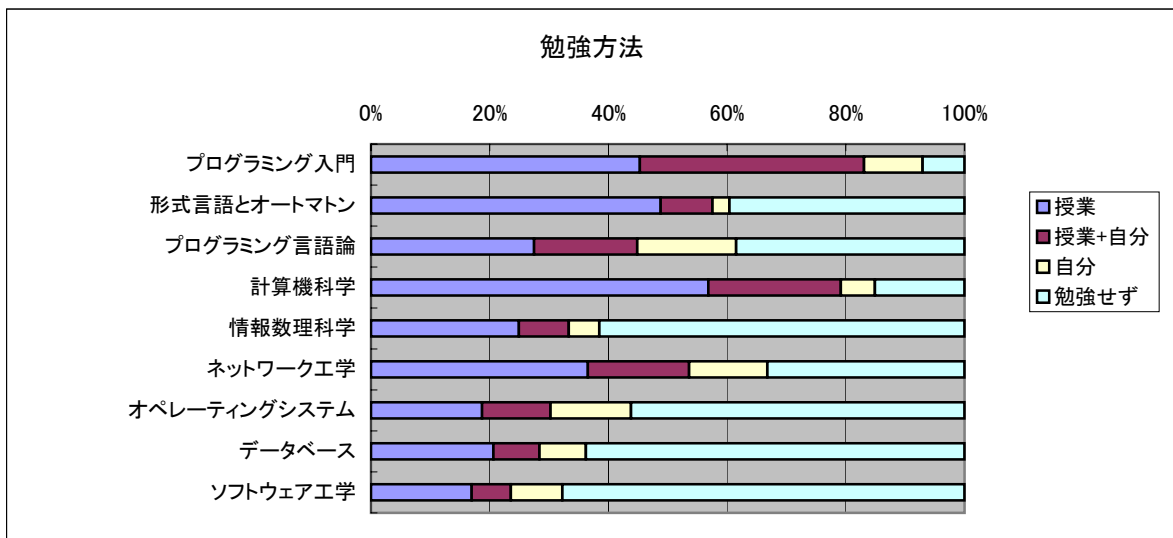
2-1-c-9. 勉強方法と効果：ソフトウェア工学

※ 参考：アンケート項目の科目分類と科目の関係

科目分類	科目	科目分類	科目
プログラミング入門	- プログラミング入門	オペレーティングシステム	- オペレーティングシステム コンパイラの役割と働き
形式言語とオートマトン	- 形式言語とオートマトン		UNIX LINUX WINDOWS MAC OS TRON Main Frame OS その他
プログラミング言語論	- プログラミング言語論 プログラミング言語体系 汎用系 インタプリタ言語 C言語 オブジェクト指向言語 スクリプト系言語 Web系言語	データベース	- データベース データベースの原理・理論 データの分析と設計方法 データ操作 DBMSの機能と特徴 分散データベース データベース設計論 汎用機のDB リレーショナルDB オブジェクト指向DB その他
計算機科学	- 計算機科学 基礎理論 データ構造 アルゴリズム 計算機アーキテクチャ	ソフトウェア工学	- ソフトウェア工学 ソフトウェア記述言語 ソフトウェア開発プロセス プロジェクトマネジメント ソフトウェア要求定義 ソフトウェア設計方法論 ソフトウェア構築方法論 ソフトウェア品質 ソフトウェアテスト ソフトウェア保守 ソフトウェア構成管理 ツール及び手法
情報数理学	- 情報数理学 多次元データ解析 パターン認識 マルチメディア AI 機械学習 最適化手法 ファジー カオス エージェント データマイニング		
ネットワーク工学	- ネットワーク工学 プロトコルと伝送制御 ネットワークアーキテクチャ 伝送制御方式 符号化と伝送 LANとWAN インターネット応用		

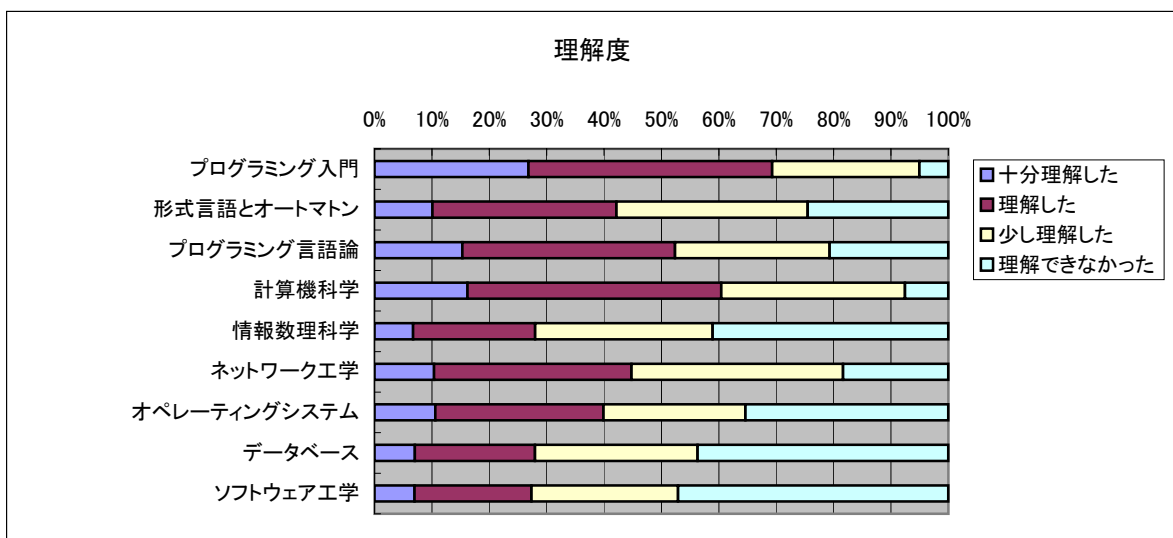
## 2-1-a-1. 勉強方法(全体) ※科目分類毎に当該配下の全科目の数値を集計

科目分類	授業	授業+自分	自分	勉強せず
プログラミング入門	289	241	63	45
形式言語とオートマトン	308	55	18	250
プログラミング言語論	1344	851	815	1883
計算機科学	1702	669	172	452
情報数理学	1621	548	336	4004
ネットワーク工学	1507	705	543	1371
オペレーティングシステム	922	569	669	2771
データベース	1216	458	462	3761
ソフトウェア工学	1192	464	609	4760



## 2-1-a-2. 理解度(全体) ※科目分類毎に当該配下の全科目の数値を集計

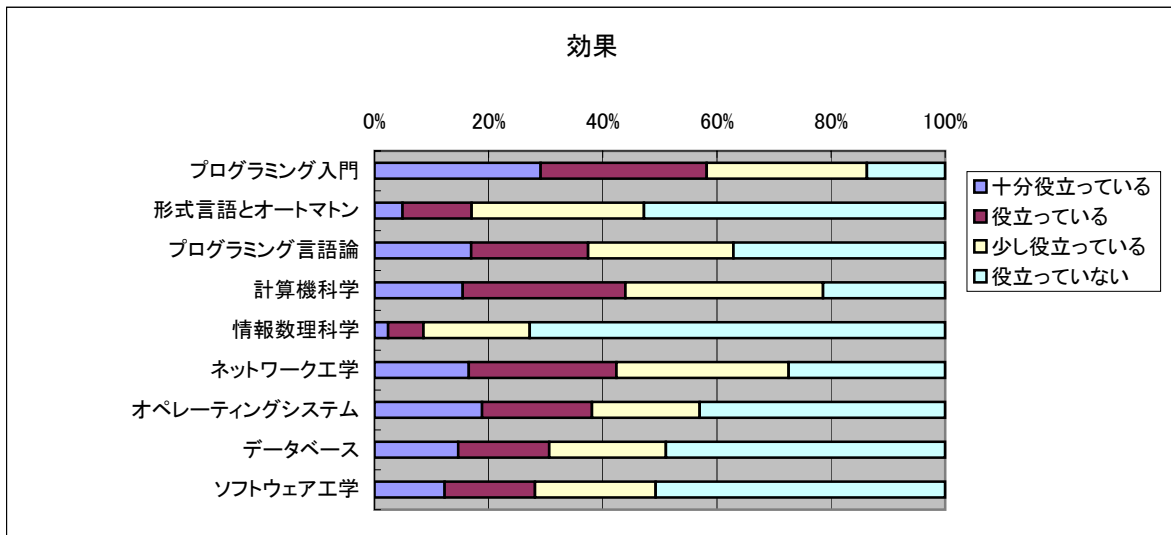
科目分類	十分理解した	理解した	少し理解した	理解できなかった
プログラミング入門	159	252	152	30
形式言語とオートマトン	47	149	155	114
プログラミング言語論	550	1332	968	744
計算機科学	422	1152	834	197
情報数理学	263	832	1210	1606
ネットワーク工学	325	1077	1154	574
オペレーティングシステム	335	926	782	1117
データベース	248	741	1002	1545
ソフトウェア工学	280	817	1024	1889



## 2-1-a-3. 効果(全体)

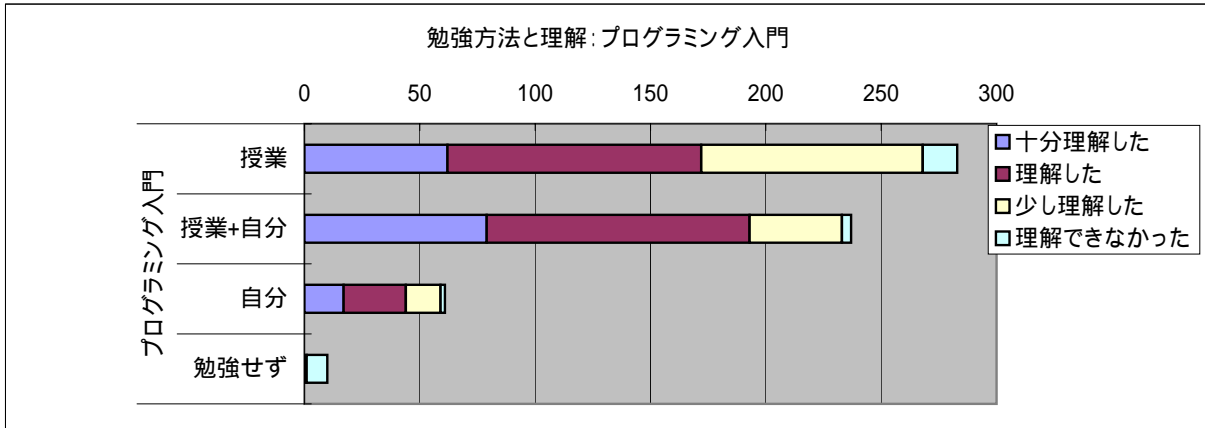
※科目分類毎に当該配下の全科目の数値を集計

科目分類	十分役立っている	役立つ	少し役立っている	役立っていない
プログラミング入門	172	172	166	81
形式言語とオートマトン	23	57	142	248
プログラミング言語論	603	729	907	1320
計算機科学	395	730	886	547
情報数理論	94	243	730	2857
ネットワーク工学	513	805	938	852
オペレーティングシステム	591	603	592	1348
データベース	520	566	724	1735
ソフトウェア工学	491	634	845	2029



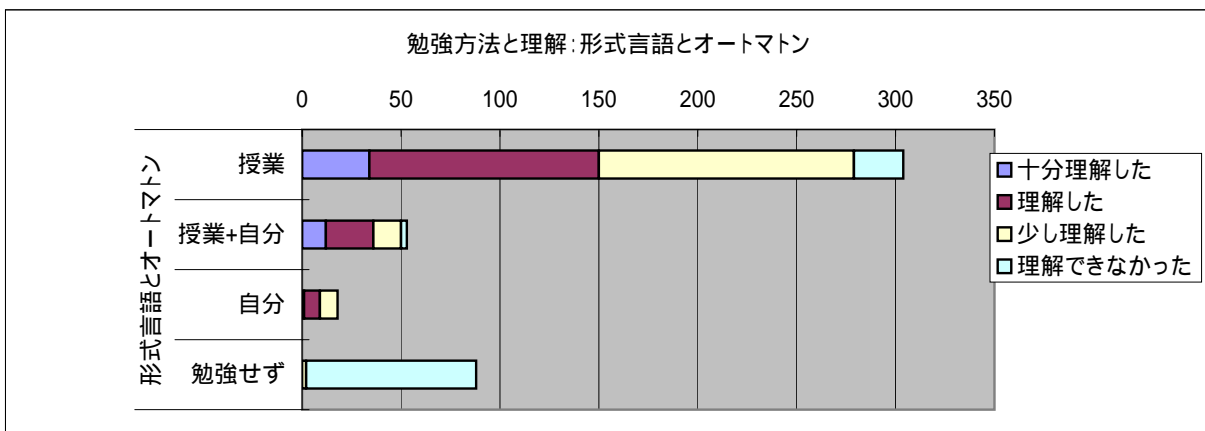
## 2 - 1 - b - 1 . 勉強方法と理解: プログラミング入門

		十分理解した	理解した	少し理解した	理解できなかった
プログラミング入門	授業	62	110	96	15
	授業+自分	79	114	40	4
	自分	17	27	15	2
	勉強せず	1			9



## 2 - 1 - b - 2 . 勉強方法と理解: 形式言語とオートマトン

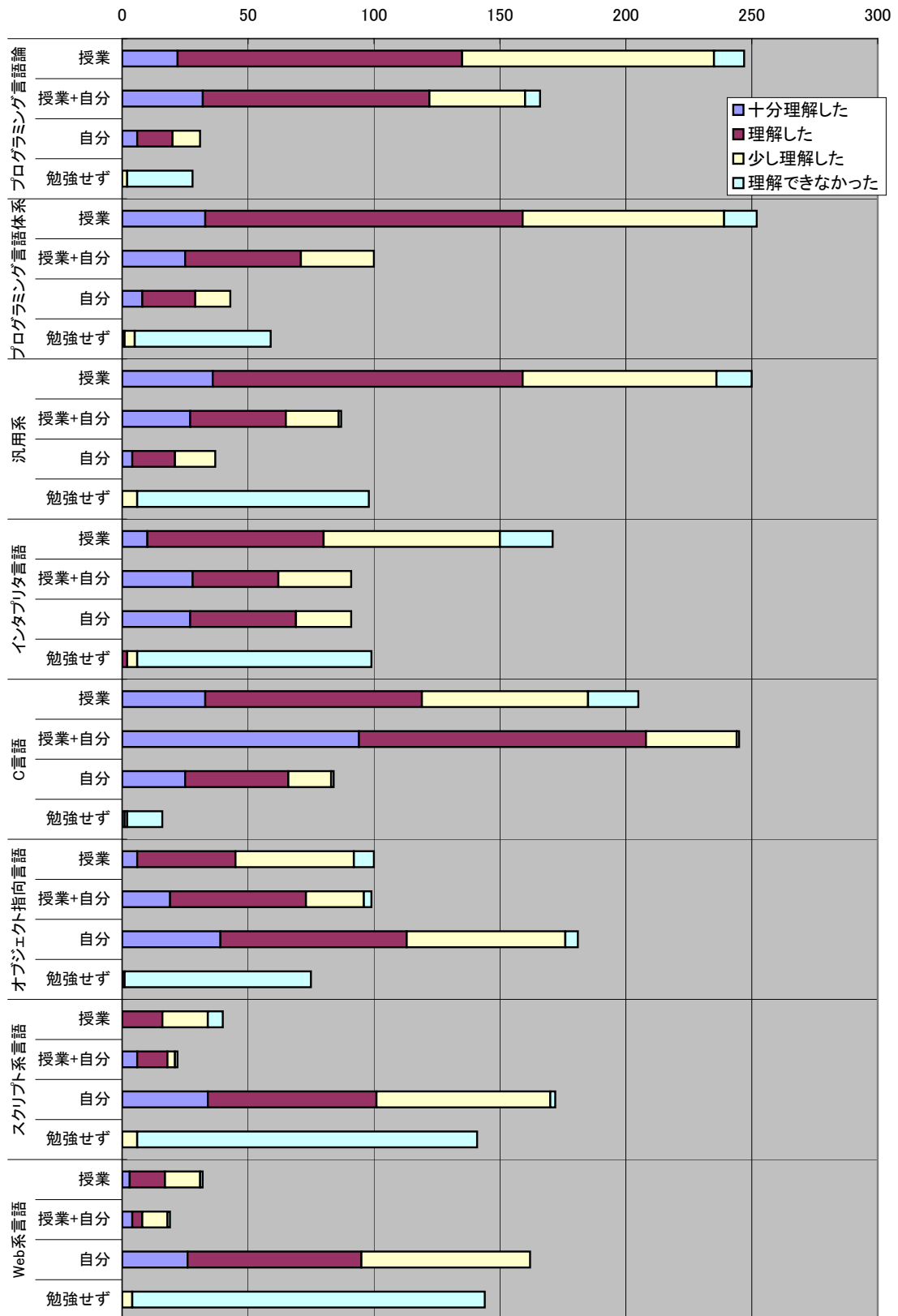
		十分理解した	理解した	少し理解した	理解できなかった
形式言語とオートマトン	授業	34	116	129	25
	授業+自分	12	24	14	3
	自分	1	8	9	2
	勉強せず	0	0	2	86



## 2 - 1 - b - 3 . 勉強方法と理解:プログラミング言語論

		十分理解した	理解した	少し理解した	理解できなかった
プログラミング言語論	授業	22	113	100	12
	授業+自分	32	90	38	6
	自分	6	14	11	
	勉強せず			2	26
プログラミング言語体系	授業	33	126	80	13
	授業+自分	25	46	29	
	自分	8	21	14	
	勉強せず		1	4	54
汎用系	授業	36	123	77	14
	授業+自分	27	38	21	1
	自分	4	17	16	
	勉強せず			6	92
インタプリタ言語	授業	10	70	70	21
	授業+自分	28	34	29	
	自分	27	42	22	
	勉強せず		2	4	93
C言語	授業	33	86	66	20
	授業+自分	94	114	36	1
	自分	25	41	17	1
	勉強せず	1		1	14
オブジェクト指向言語	授業	6	39	47	8
	授業+自分	19	54	23	3
	自分	39	74	63	5
	勉強せず		1		74
スクリプト系言語	授業		16	18	6
	授業+自分	6	12	3	1
	自分	34	67	69	2
	勉強せず			6	135
Web系言語	授業	3	14	14	1
	授業+自分	4	4	10	1
	自分	26	69	67	
	勉強せず			4	140

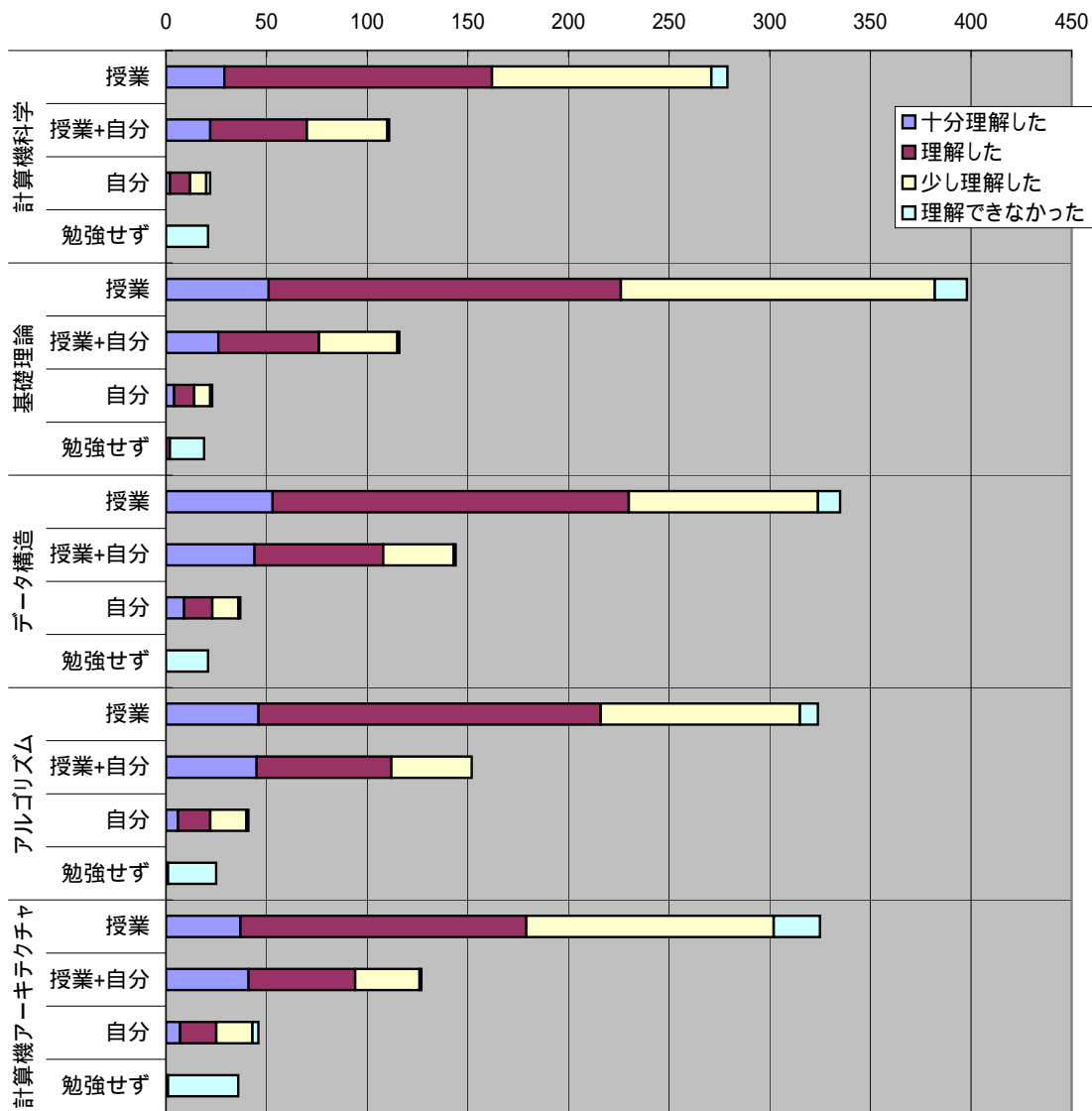
勉強方法と理解：プログラミング言語論



## 2 - 1 - b - 4 . 勉強方法と理解: 計算機科学

		十分理解した	理解した	少し理解した	理解できなかった
計算機科学	授業	29	133	109	8
	授業+自分	22	48	40	1
	自分	2	10	8	2
	勉強せず				21
基礎理論	授業	51	175	156	16
	授業+自分	26	50	39	1
	自分	4	10	8	1
	勉強せず		2		17
データ構造	授業	53	177	94	11
	授業+自分	44	64	35	1
	自分	9	14	13	1
	勉強せず				21
アルゴリズム	授業	46	170	99	9
	授業+自分	45	67	40	
	自分	6	16	18	1
	勉強せず			1	24
計算機アーキテクチャ	授業	37	142	123	23
	授業+自分	41	53	32	1
	自分	7	18	18	3
	勉強せず		1		35

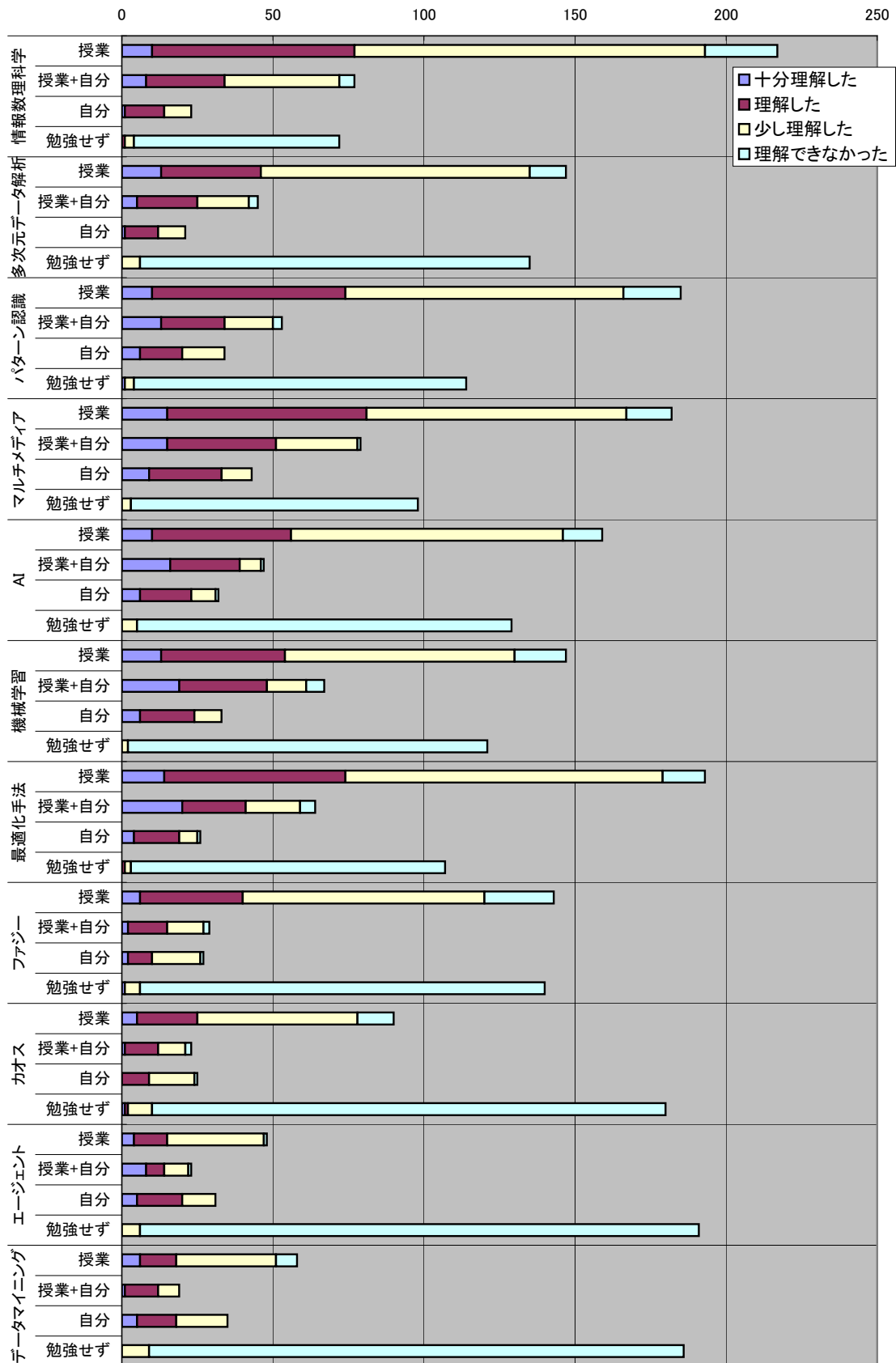
勉強方法と理解: 計算機科学



## 2 - 1 - b - 5 . 勉強方法と理解:情報数理学

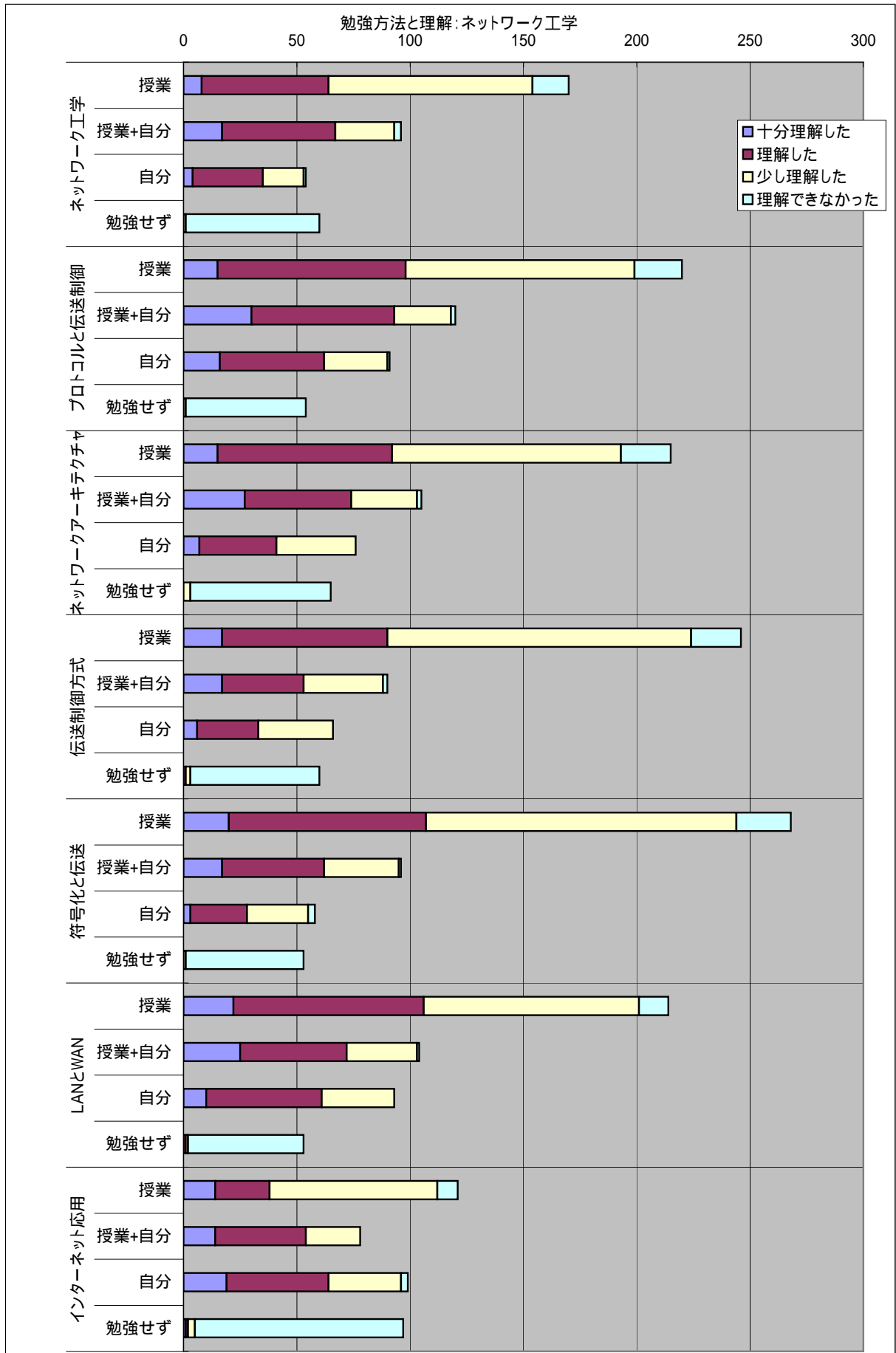
		十分理解した	理解した	少し理解した	理解できなかった
情報数理学	授業	10	67	116	24
	授業+自分	8	26	38	5
	自分	1	13	9	
	勉強せず		1	3	68
多次元データ解析	授業	13	33	89	12
	授業+自分	5	20	17	3
	自分	1	11	9	
	勉強せず			6	129
パターン認識	授業	10	64	92	19
	授業+自分	13	21	16	3
	自分	6	14	14	
	勉強せず	1		3	110
マルチメディア	授業	15	66	86	15
	授業+自分	15	36	27	1
	自分	9	24	10	
	勉強せず			3	95
AI	授業	10	46	90	13
	授業+自分	16	23	7	1
	自分	6	17	8	1
	勉強せず			5	124
機械学習	授業	13	41	76	17
	授業+自分	19	29	13	6
	自分	6	18	9	
	勉強せず			2	119
最適化手法	授業	14	60	105	14
	授業+自分	20	21	18	5
	自分	4	15	6	1
	勉強せず		1	2	104
ファジー	授業	6	34	80	23
	授業+自分	2	13	12	2
	自分	2	8	16	1
	勉強せず	1		5	134
カオス	授業	5	20	53	12
	授業+自分	1	11	9	2
	自分		9	15	1
	勉強せず	1	1	8	170
エージェント	授業	4	11	32	1
	授業+自分	8	6	8	1
	自分	5	15	11	
	勉強せず			6	185
データマイニング	授業	6	12	33	7
	授業+自分	1	11	7	
	自分	5	13	17	
	勉強せず			9	177

勉強方法と理解:情報数理学



## 2 - 1 - b - 6 . 勉強方法と理解: ネットワーク工学

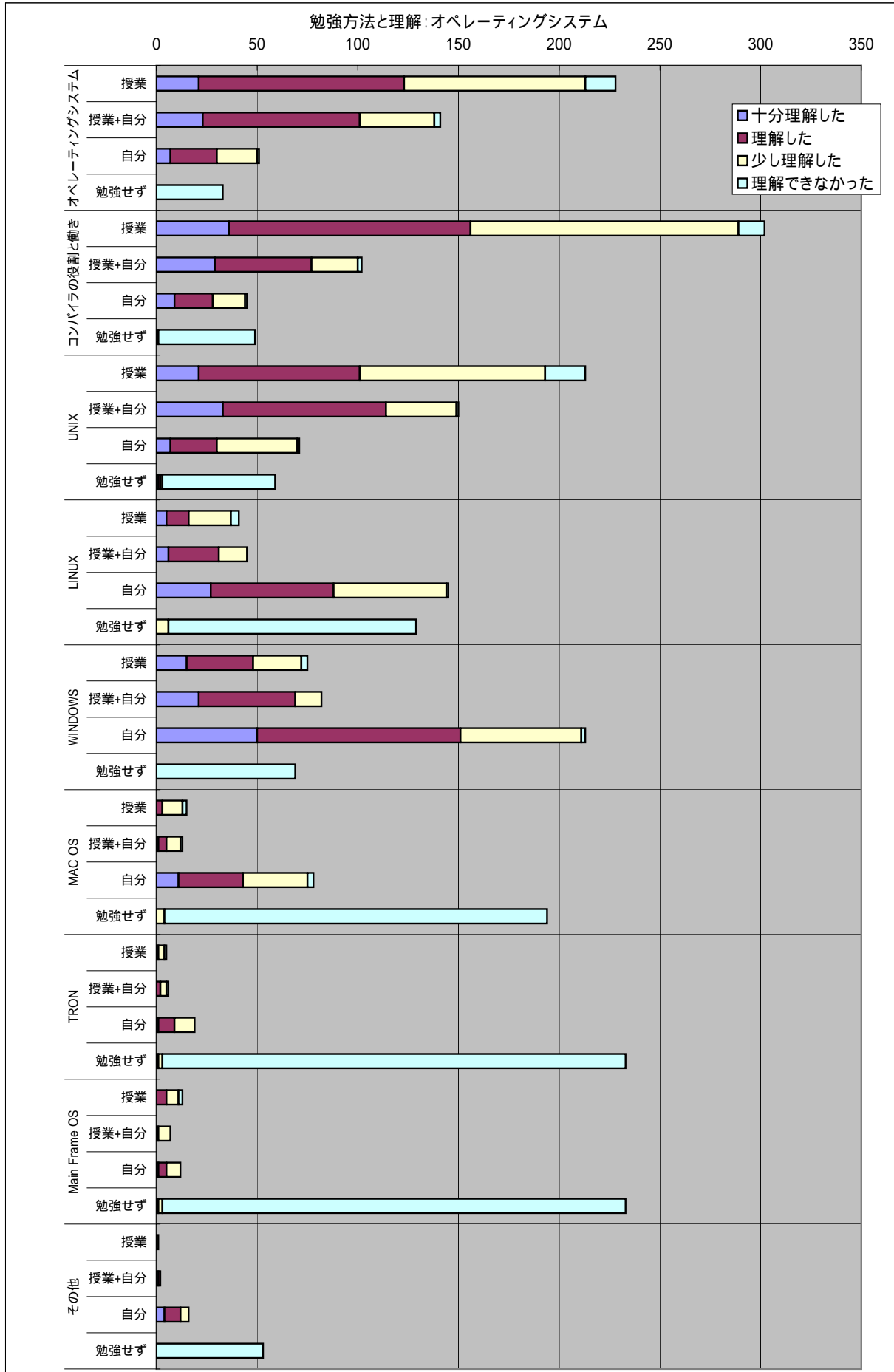
		十分理解した	理解した	少し理解した	理解できなかった
ネットワーク工学	授業	8	56	90	16
	授業+自分	17	50	26	3
	自分	4	31	18	1
	勉強せず			1	59
プロトコルと伝送制御	授業	15	83	101	21
	授業+自分	30	63	25	2
	自分	16	46	28	1
	勉強せず			1	53
ネットワークアーキテクチャ	授業	15	77	101	22
	授業+自分	27	47	29	2
	自分	7	34	35	
	勉強せず			3	62
伝送制御方式	授業	17	73	134	22
	授業+自分	17	36	35	2
	自分	6	27	33	
	勉強せず		1	2	57
符号化と伝送	授業	20	87	137	24
	授業+自分	17	45	33	1
	自分	3	25	27	3
	勉強せず			1	52
LANとWAN	授業	22	84	95	13
	授業+自分	25	47	31	1
	自分	10	51	32	
	勉強せず		1	1	51
インターネット応用	授業	14	24	74	9
	授業+自分	14	40	24	
	自分	19	45	32	3
	勉強せず	1	1	3	92



## 2 - 1 - b - 7 . 勉強方法と理解 : オペレーティングシステム

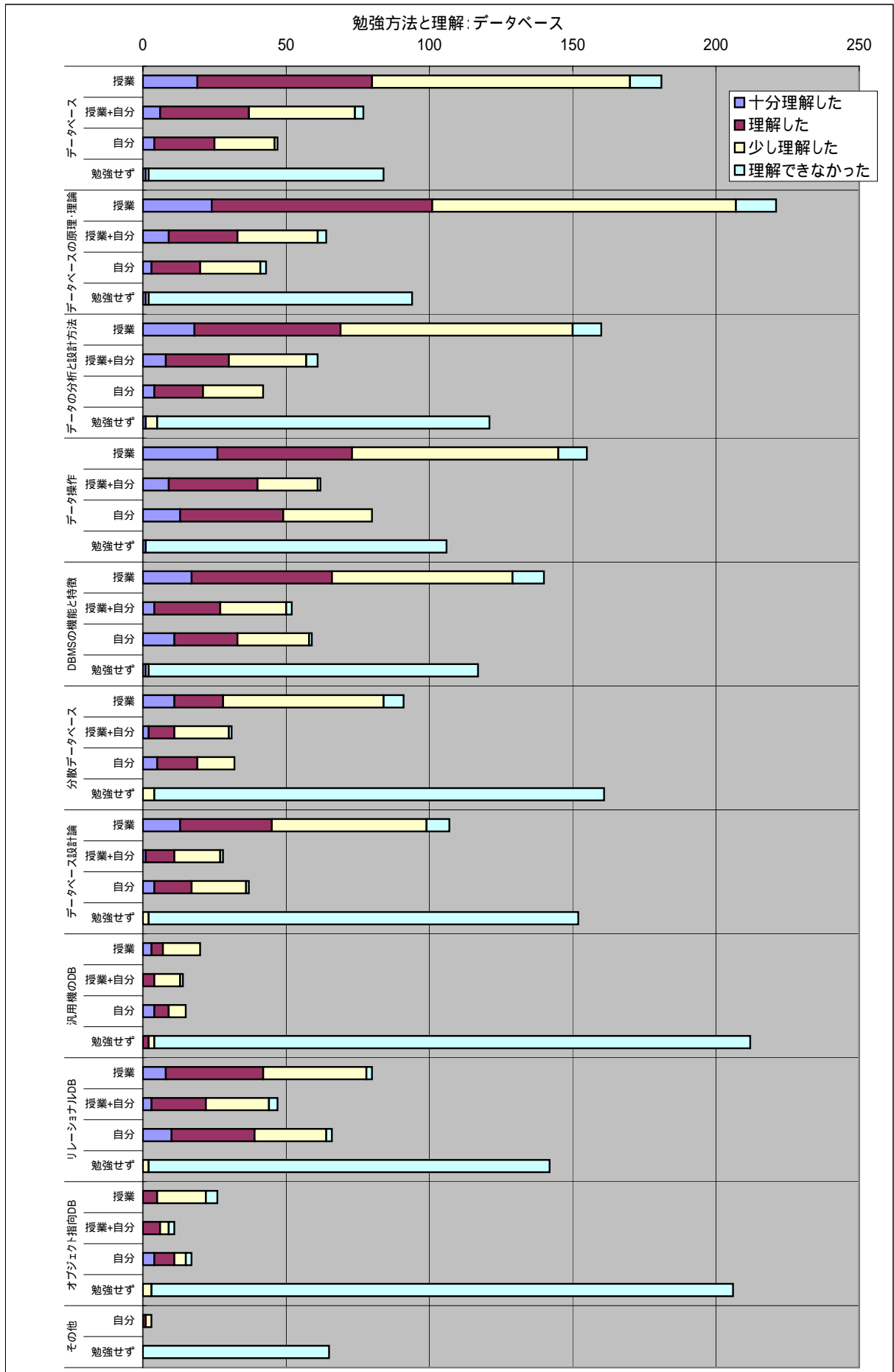
		十分理解した	理解した	少し理解した	理解できなかった
オペレーティングシステム	授業	21	102	90	15
	授業+自分	23	78	37	3
	自分	7	23	20	1
	勉強せず				33
コンパイラの役割と働き	授業	36	120	133	13
	授業+自分	29	48	23	2
	自分	9	19	16	1
	勉強せず			1	48
UNIX	授業	21	80	92	20
	授業+自分	33	81	35	1
	自分	7	23	40	1
	勉強せず	1	1	1	56
LINUX	授業	5	11	21	4
	授業+自分	6	25	14	
	自分	27	61	56	1
	勉強せず			6	123
WINDOWS	授業	15	33	24	3
	授業+自分	21	48	13	
	自分	50	101	60	2
	勉強せず				69
MAC OS	授業		3	10	2
	授業+自分	1	4	7	1
	自分	11	32	32	3
	勉強せず			4	190
TRON	授業	1		3	1
	授業+自分		2	3	1
	自分	1	8	10	
	勉強せず		1	2	230
Main Frame OS	授業		5	6	2
	授業+自分		1	6	
	自分	1	4	7	
	勉強せず		1	2	230
その他	授業			1	
	授業+自分	1	1		
	自分	4	8	4	
	勉強せず				53

勉強方法と理解:オペレーティングシステム



## 2 - 1 - b - 8 . 勉強方法と理解: データベース

		十分理解した	理解した	少し理解した	理解できなかった
データベース	授業	19	61	90	11
	授業+自分	6	31	37	3
	自分	4	21	21	1
	勉強せず	1		1	82
データベースの原理・理論	授業	24	77	106	14
	授業+自分	9	24	28	3
	自分	3	17	21	2
	勉強せず	1		1	92
データの分析と設計方法	授業	18	51	81	10
	授業+自分	8	22	27	4
	自分	4	17	21	
	勉強せず	1		4	116
データ操作	授業	26	47	72	10
	授業+自分	9	31	21	1
	自分	13	36	31	
	勉強せず	1			105
DBMSの機能と特徴	授業	17	49	63	11
	授業+自分	4	23	23	2
	自分	11	22	25	1
	勉強せず	1		1	115
分散データベース	授業	11	17	56	7
	授業+自分	2	9	19	1
	自分	5	14	13	
	勉強せず			4	157
データベース設計論	授業	13	32	54	8
	授業+自分	1	10	16	1
	自分	4	13	19	1
	勉強せず			2	150
汎用機のDB	授業	3	4	13	
	授業+自分		4	9	1
	自分	4	5	6	
	勉強せず		2	2	208
リレーショナルDB	授業	8	34	36	2
	授業+自分	3	19	22	3
	自分	10	29	25	2
	勉強せず			2	140
オブジェクト指向DB	授業		5	17	4
	授業+自分		6	3	2
	自分	4	7	4	2
	勉強せず			3	203
その他	自分		1	2	
	勉強せず				65

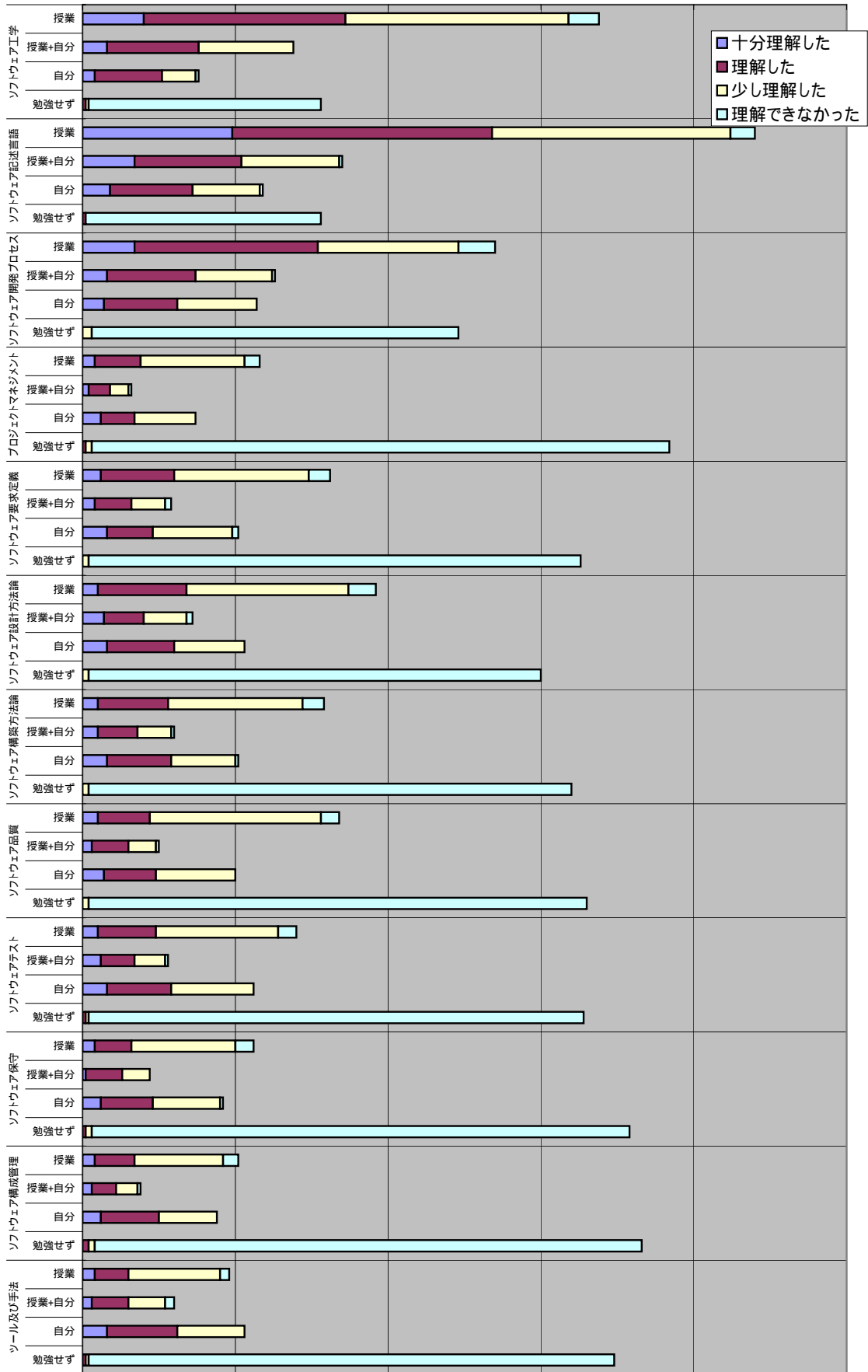


## 2 - 1 - b - 9 . 勉強方法と理解:ソフトウェア工学

		十分理解した	理解した	少し理解した	理解できなかった
ソフトウェア工学	授業	20	66	73	10
	授業+自分	8	30	31	
	自分	4	22	11	1
	勉強せず		1	1	76
ソフトウェア記述言語	授業	49	85	78	8
	授業+自分	17	35	32	1
	自分	9	27	22	1
	勉強せず		1		77
ソフトウェア開発プロセス	授業	17	60	46	12
	授業+自分	8	29	25	1
	自分	7	24	26	
	勉強せず			3	120
プロジェクトマネジメント	授業	4	15	34	5
	授業+自分	2	7	6	1
	自分	6	11	20	
	勉強せず		1	2	189
ソフトウェア要求定義	授業	6	24	44	7
	授業+自分	4	12	11	2
	自分	8	15	26	2
	勉強せず			2	161
ソフトウェア設計方法論	授業	5	29	53	9
	授業+自分	7	13	14	2
	自分	8	22	23	
	勉強せず			2	148
ソフトウェア構築方法論	授業	5	23	44	7
	授業+自分	5	13	11	1
	自分	8	21	21	1
	勉強せず			2	158
ソフトウェア品質	授業	5	17	56	6
	授業+自分	3	12	9	1
	自分	7	17	26	
	勉強せず			2	163
ソフトウェアテスト	授業	5	19	40	6
	授業+自分	6	11	10	1
	自分	8	21	27	
	勉強せず		1	1	162
ソフトウェア保守	授業	4	12	34	6
	授業+自分	1	12	9	
	自分	6	17	22	1
	勉強せず		1	2	176
ソフトウェア構成管理	授業	4	13	29	5
	授業+自分	3	8	7	1
	自分	6	19	19	
	勉強せず		2	2	179
ツール及び手法	授業	4	11	30	3
	授業+自分	3	12	12	3
	自分	8	23	22	
	勉強せず		1	1	172

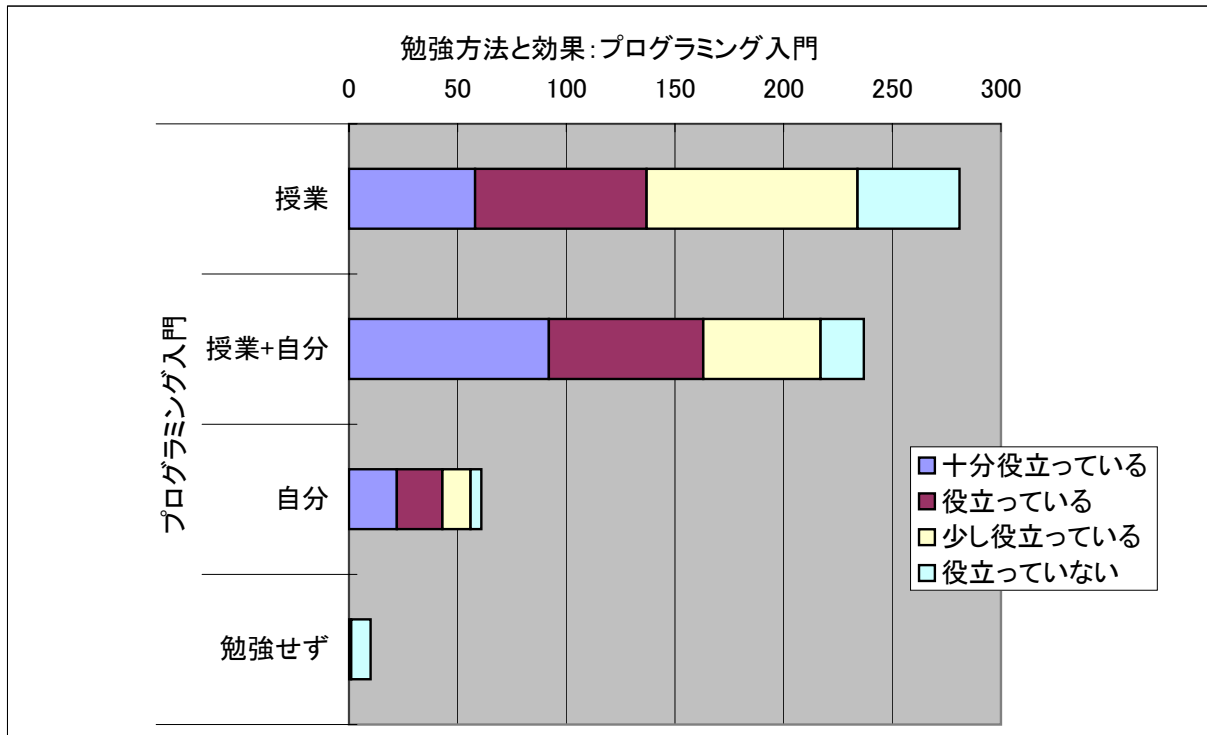
勉強方法と理解:ソフトウェア工学

0 50 100 150 200 250



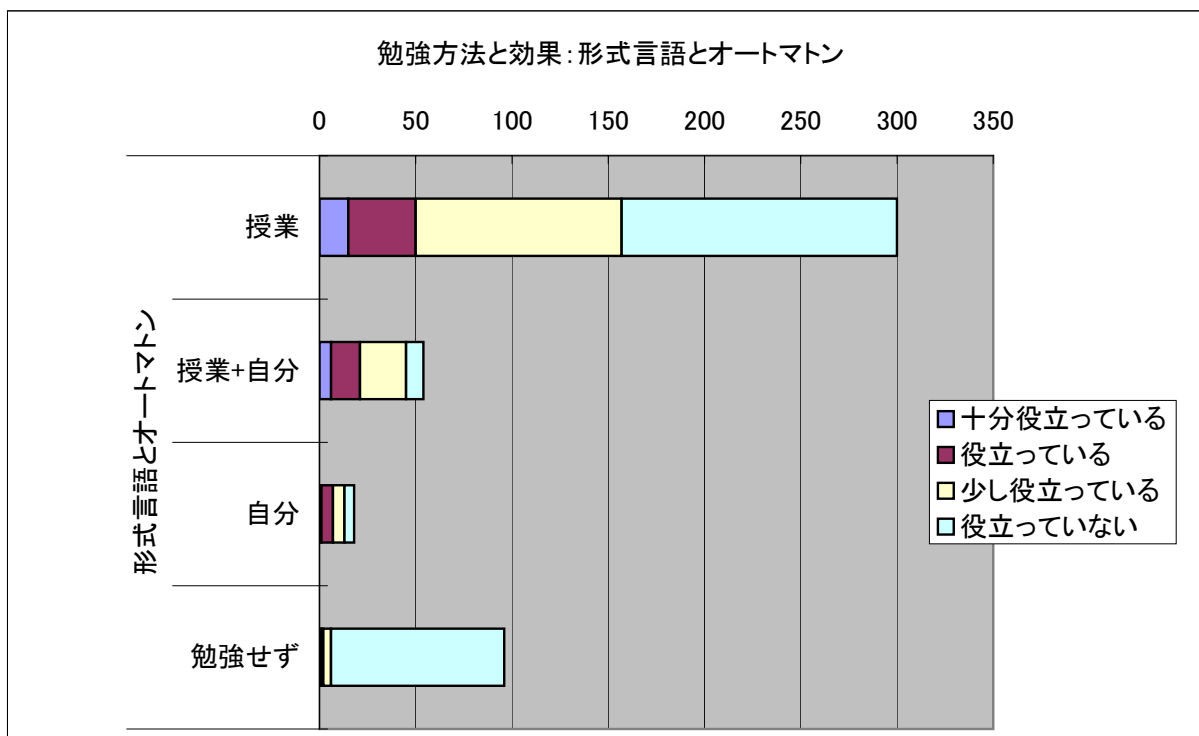
## 2-1-c-1. 勉強方法と効果:プログラミング入門

		十分役立っている	役立っている	少し役立っている	役立っていない
プログラミング入門	授業	58	79	97	47
	授業+自分	92	71	54	20
	自分	22	21	13	5
	勉強せず			1	9



## 2-1-c-2. 勉強方法と効果: 形式言語とオートマトン

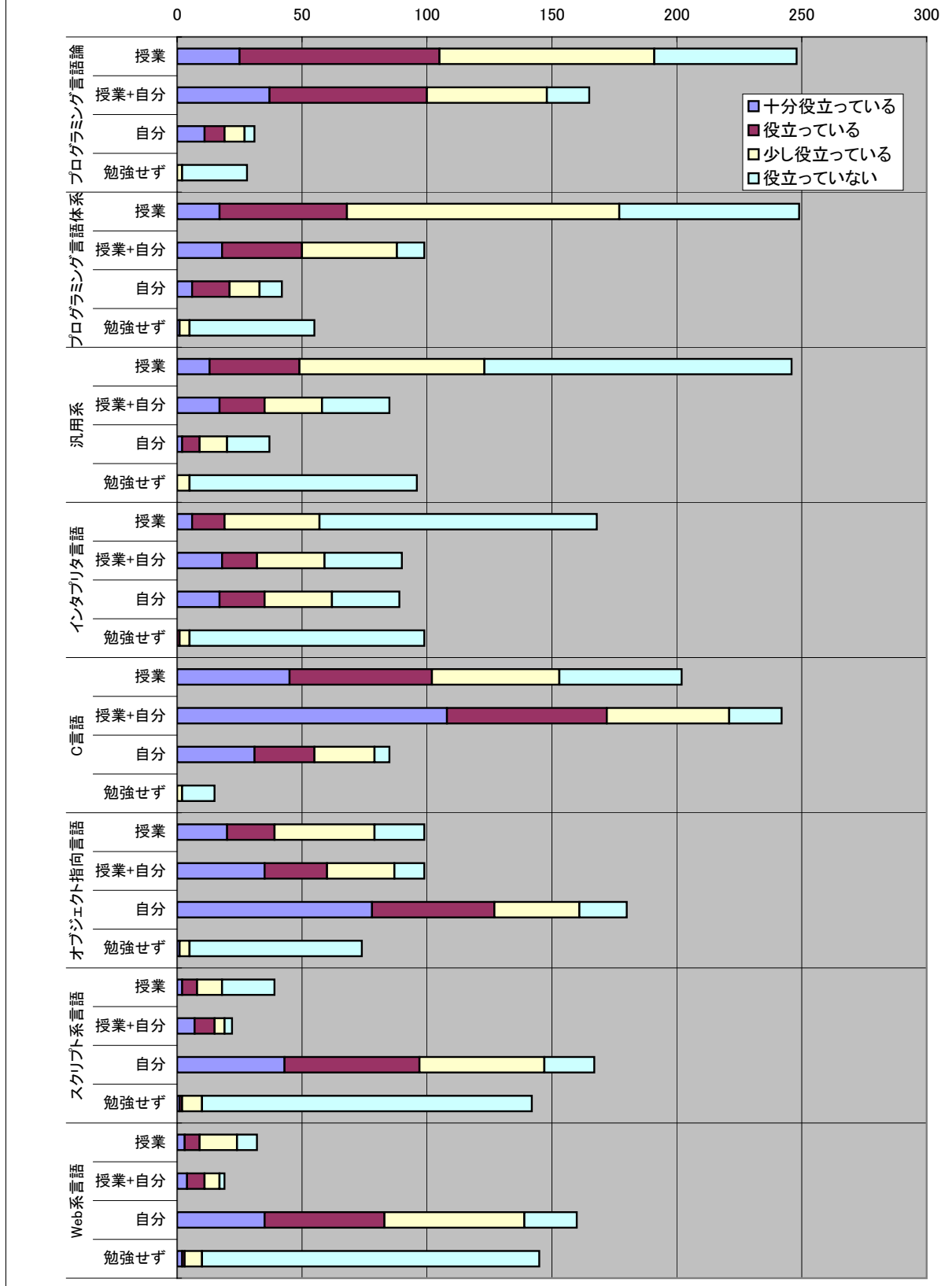
		十分役立っている	役立っている	少し役立っている	役立っていない
形式言語とオートマトン	授業	15	35	107	143
	授業+自分	6	15	24	9
	自分	1	6	6	5
	勉強せず	1	1	4	90



## 2 - 1 - c - 3 . 勉強方法と効果 : プログラミング言語論

		十分役立っている	役立っている	少し役立っている	役立っていない
プログラミング言語論	授業	25	80	86	57
	授業+自分	37	63	48	17
	自分	11	8	8	4
	勉強せず			2	26
プログラミング言語体系	授業	17	51	109	72
	授業+自分	18	32	38	11
	自分	6	15	12	9
	勉強せず	1		4	50
汎用系	授業	13	36	74	123
	授業+自分	17	18	23	27
	自分	2	7	11	17
	勉強せず			5	91
インタプリタ言語	授業	6	13	38	111
	授業+自分	18	14	27	31
	自分	17	18	27	27
	勉強せず		1	4	94
C言語	授業	45	57	51	49
	授業+自分	108	64	49	21
	自分	31	24	24	6
	勉強せず			2	13
オブジェクト指向言語	授業	20	19	40	20
	授業+自分	35	25	27	12
	自分	78	49	34	19
	勉強せず	1		4	69
スクリプト系言語	授業	2	6	10	21
	授業+自分	7	8	4	3
	自分	43	54	50	20
	勉強せず	1	1	8	132
Web系言語	授業	3	6	15	8
	授業+自分	4	7	6	2
	自分	35	48	56	21
	勉強せず	2	1	7	135

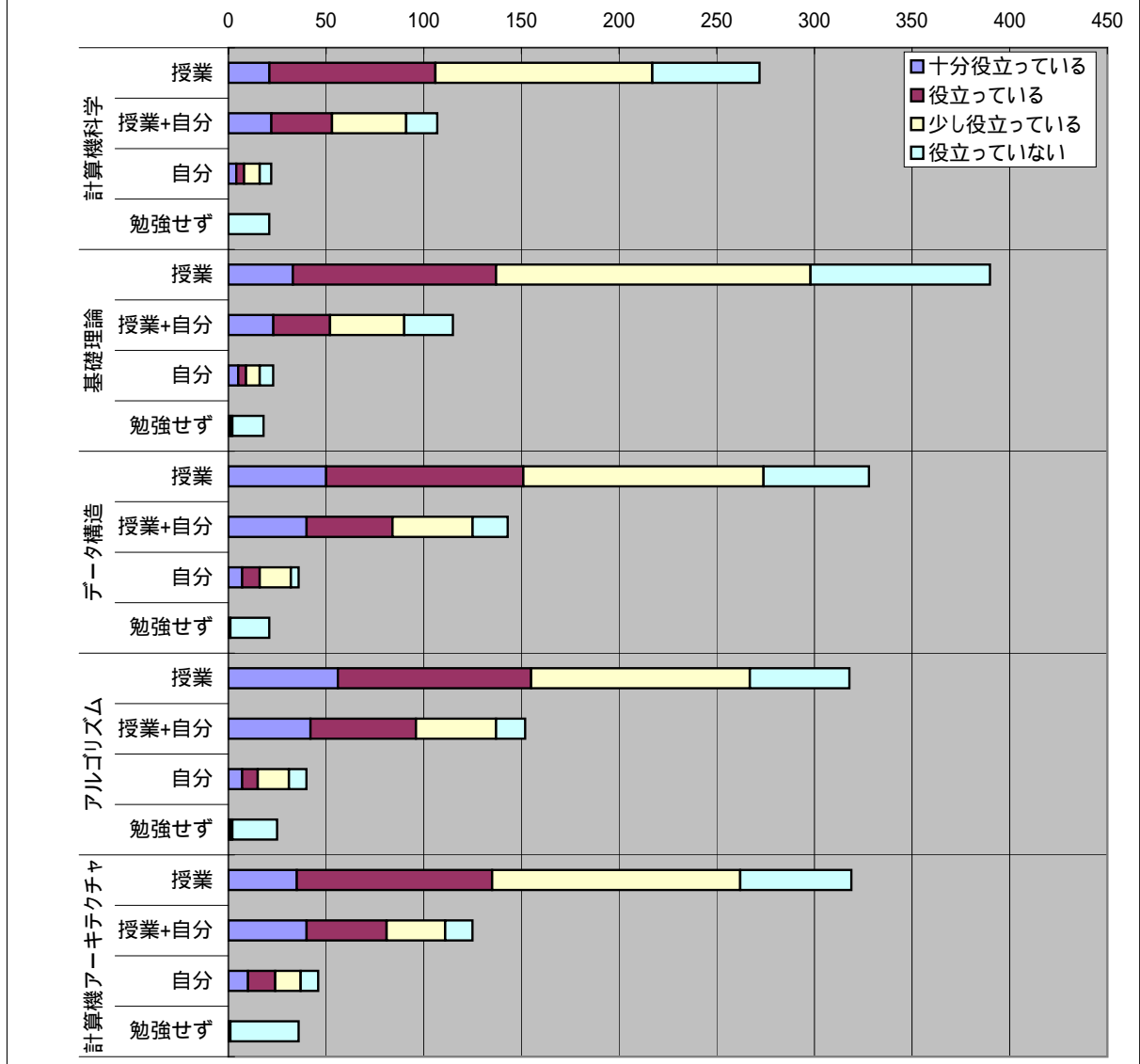
勉強方法と効果:プログラミング言語論



## 2 - 1 - c - 4 . 勉強方法と効果 : 計算機科学

		十分役立っている	役立っている	少し役立っている	役立っていない
計算機科学	授業	21	85	111	55
	授業+自分	22	31	38	16
	自分	4	4	8	6
	勉強せず				21
基礎理論	授業	33	104	161	92
	授業+自分	23	29	38	25
	自分	5	4	7	7
	勉強せず		1	1	16
データ構造	授業	50	101	123	54
	授業+自分	40	44	41	18
	自分	7	9	16	4
	勉強せず		1		20
アルゴリズム	授業	56	99	112	51
	授業+自分	42	54	41	15
	自分	7	8	16	9
	勉強せず		1	1	23
計算機アーキテクチャ	授業	35	100	127	57
	授業+自分	40	41	30	14
	自分	10	14	13	9
	勉強せず			1	35

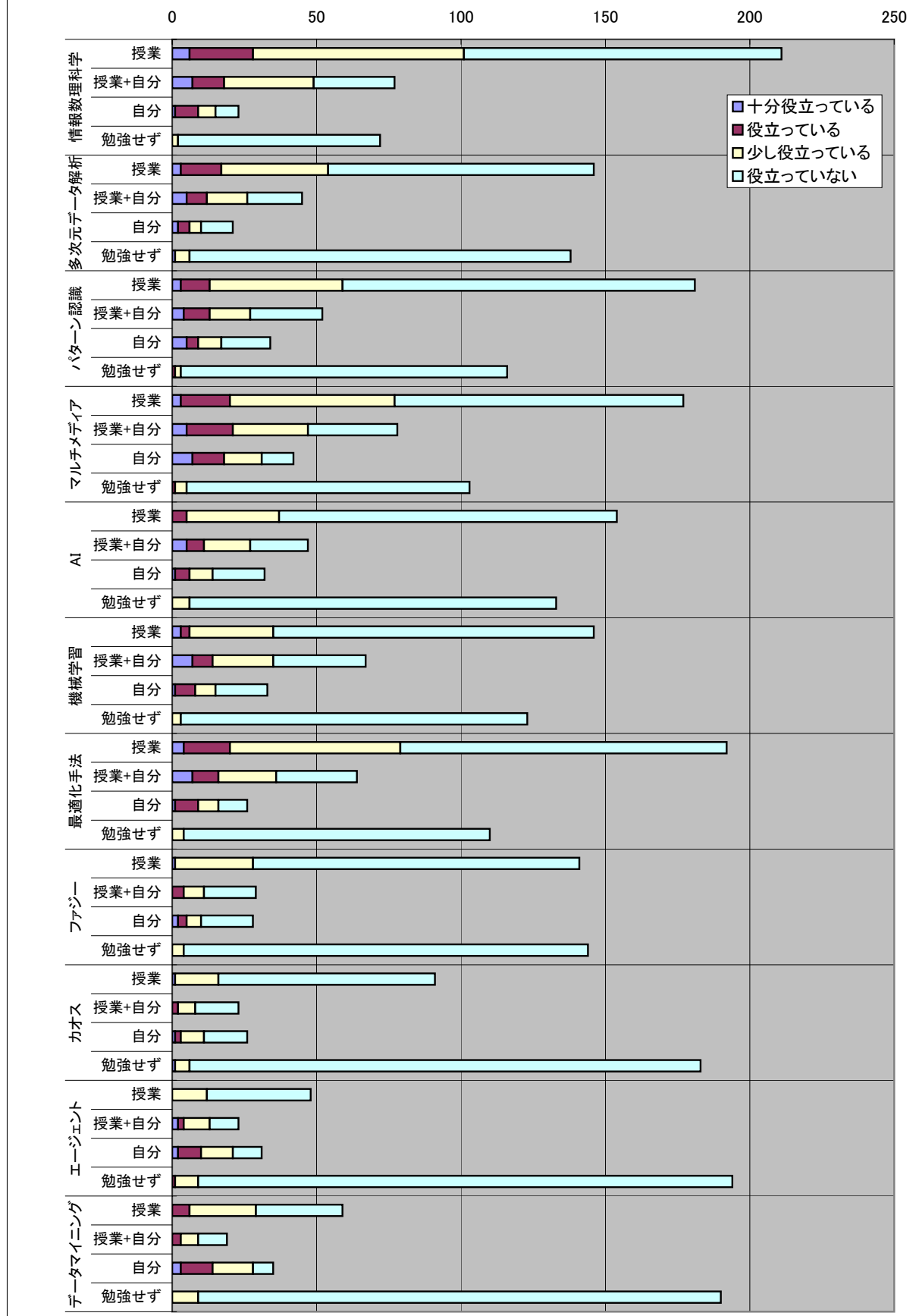
勉強方法と効果: 計算機科学



## 2 - 1 - c - 5 . 勉強方法と効果 : 情報数理学

		十分役立っている	役立っている	少し役立っている	役立っていない
情報数理学	授業	6	22	73	110
	授業+自分	7	11	31	28
	自分	1	8	6	8
	勉強せず			2	70
多次元データ解析	授業	3	14	37	92
	授業+自分	5	7	14	19
	自分	2	4	4	11
	勉強せず	1		5	132
パターン認識	授業	3	10	46	122
	授業+自分	4	9	14	25
	自分	5	4	8	17
	勉強せず		1	2	113
マルチメディア	授業	3	17	57	100
	授業+自分	5	16	26	31
	自分	7	11	13	11
	勉強せず		1	4	98
AI	授業		5	32	117
	授業+自分	5	6	16	20
	自分	1	5	8	18
	勉強せず			6	127
機械学習	授業	3	3	29	111
	授業+自分	7	7	21	32
	自分	1	7	7	18
	勉強せず			3	120
最適化手法	授業	4	16	59	113
	授業+自分	7	9	20	28
	自分	1	8	7	10
	勉強せず			4	106
ファジー	授業	1		27	113
	授業+自分		4	7	18
	自分	2	3	5	18
	勉強せず			4	140
カオス	授業	1		15	75
	授業+自分		2	6	15
	自分	1	2	8	15
	勉強せず	1		5	177
エージェント	授業			12	36
	授業+自分	2	2	9	10
	自分	2	8	11	10
	勉強せず		1	8	185
データマイニング	授業		6	23	30
	授業+自分		3	6	10
	自分	3	11	14	7
	勉強せず			9	181

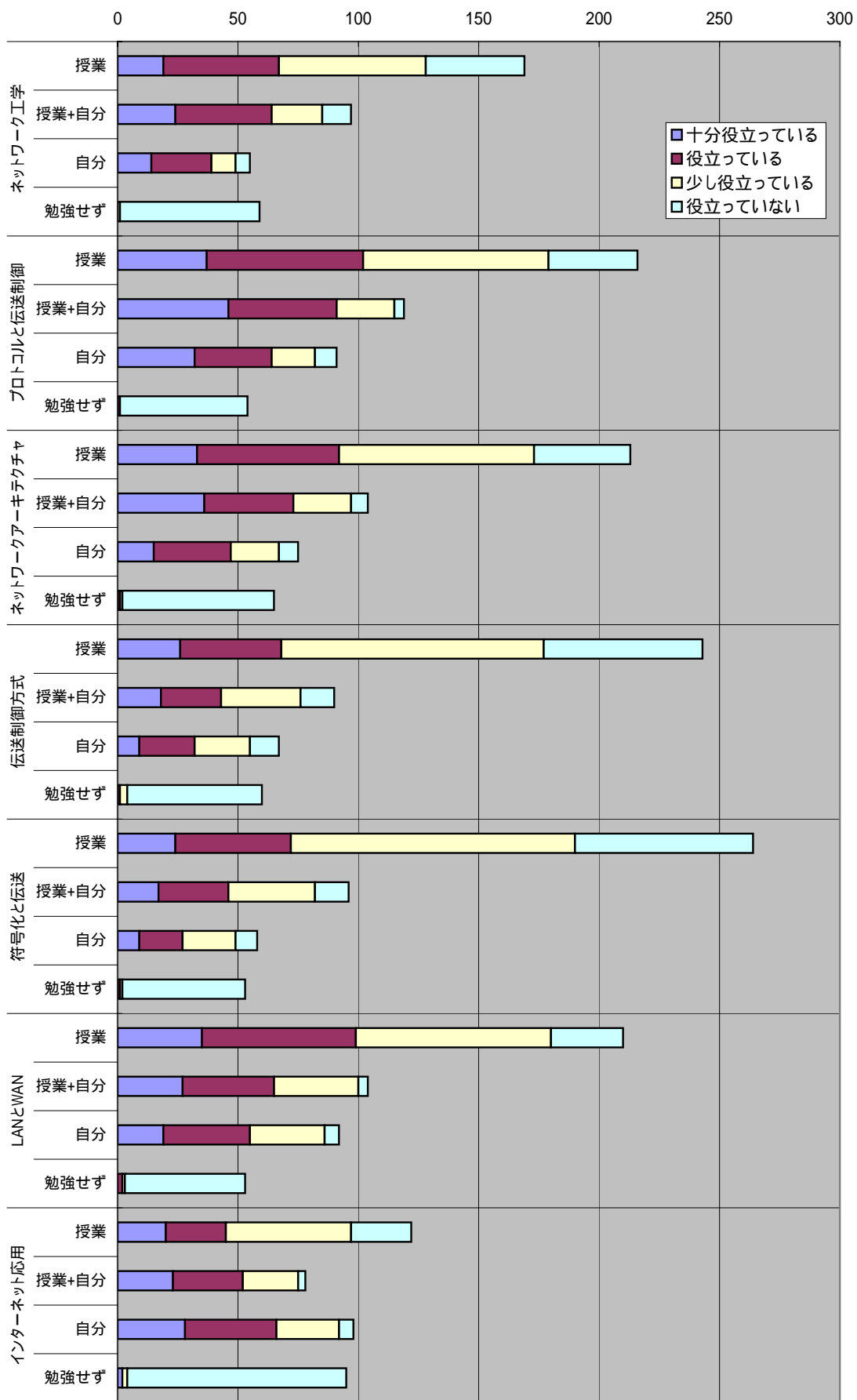
勉強方法と効果:情報数理学



## 2 - 1 - c - 6 . 勉強方法と効果 : ネットワーク工学

		十分役立っている	役立っている	少し役立っている	役立っていない
ネットワーク工学	授業	19	48	61	41
	授業+自分	24	40	21	12
	自分	14	25	10	6
	勉強せず			1	58
プロトコルと伝送制御	授業	37	65	77	37
	授業+自分	46	45	24	4
	自分	32	32	18	9
	勉強せず		1		53
ネットワークアーキテクチャ	授業	33	59	81	40
	授業+自分	36	37	24	7
	自分	15	32	20	8
	勉強せず		1	1	63
伝送制御方式	授業	26	42	109	66
	授業+自分	18	25	33	14
	自分	9	23	23	12
	勉強せず		1	3	56
符号化と伝送	授業	24	48	118	74
	授業+自分	17	29	36	14
	自分	9	18	22	9
	勉強せず		1	1	51
LANとWAN	授業	35	64	81	30
	授業+自分	27	38	35	4
	自分	19	36	31	6
	勉強せず		2	1	50
インターネット応用	授業	20	25	52	25
	授業+自分	23	29	23	3
	自分	28	38	26	6
	勉強せず	2		2	91

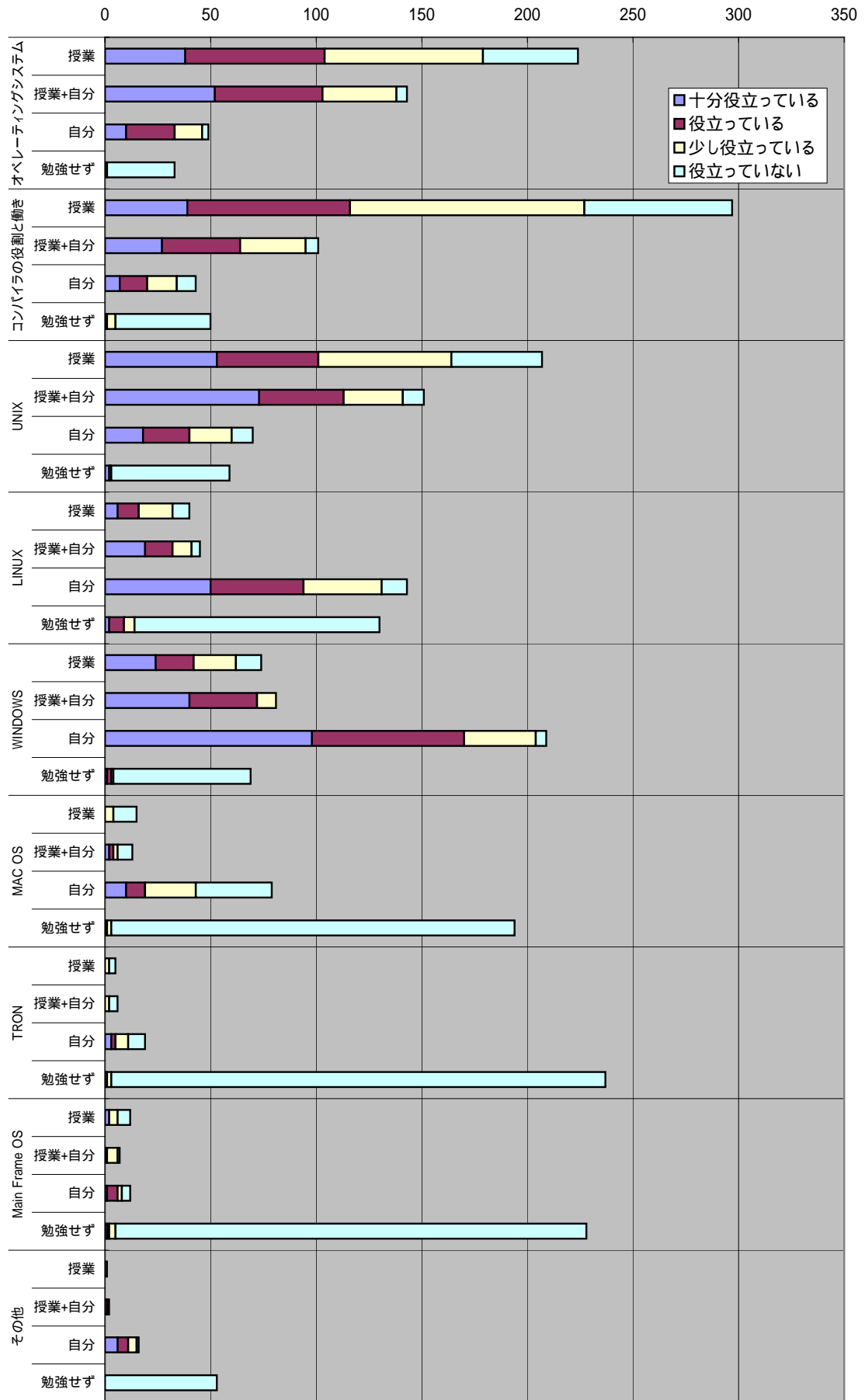
勉強方法と効果:ネットワーク工学



## 2 - 1 - c - 7 . 勉強方法と効果 : オペレーティングシステム

		十分役立っている	役立っている	少し役立っている	役立っていない
オペレーティングシステム	授業	38	66	75	45
	授業+自分	52	51	35	5
	自分	10	23	13	3
	勉強せず			1	32
コンパイラの役割と働き	授業	39	77	111	70
	授業+自分	27	37	31	6
	自分	7	13	14	9
	勉強せず		1	4	45
UNIX	授業	53	48	63	43
	授業+自分	73	40	28	10
	自分	18	22	20	10
	勉強せず	2		1	56
LINUX	授業	6	10	16	8
	授業+自分	19	13	9	4
	自分	50	44	37	12
	勉強せず	2	7	5	116
WINDOWS	授業	24	18	20	12
	授業+自分	40	32	9	
	自分	98	72	34	5
	勉強せず	1	2	1	65
MAC OS	授業			4	11
	授業+自分	2	2	2	7
	自分	10	9	24	36
	勉強せず		1	2	191
TRON	授業			2	3
	授業+自分			2	4
	自分	3	2	6	8
	勉強せず	1		2	234
Main Frame OS	授業	2		4	6
	授業+自分		1	5	1
	自分	1	5	2	4
	勉強せず	1	1	3	223
その他	授業			1	
	授業+自分	1	1		
	自分	6	5	4	1
	勉強せず				53

勉強方法と効果:オペレーティングシステム

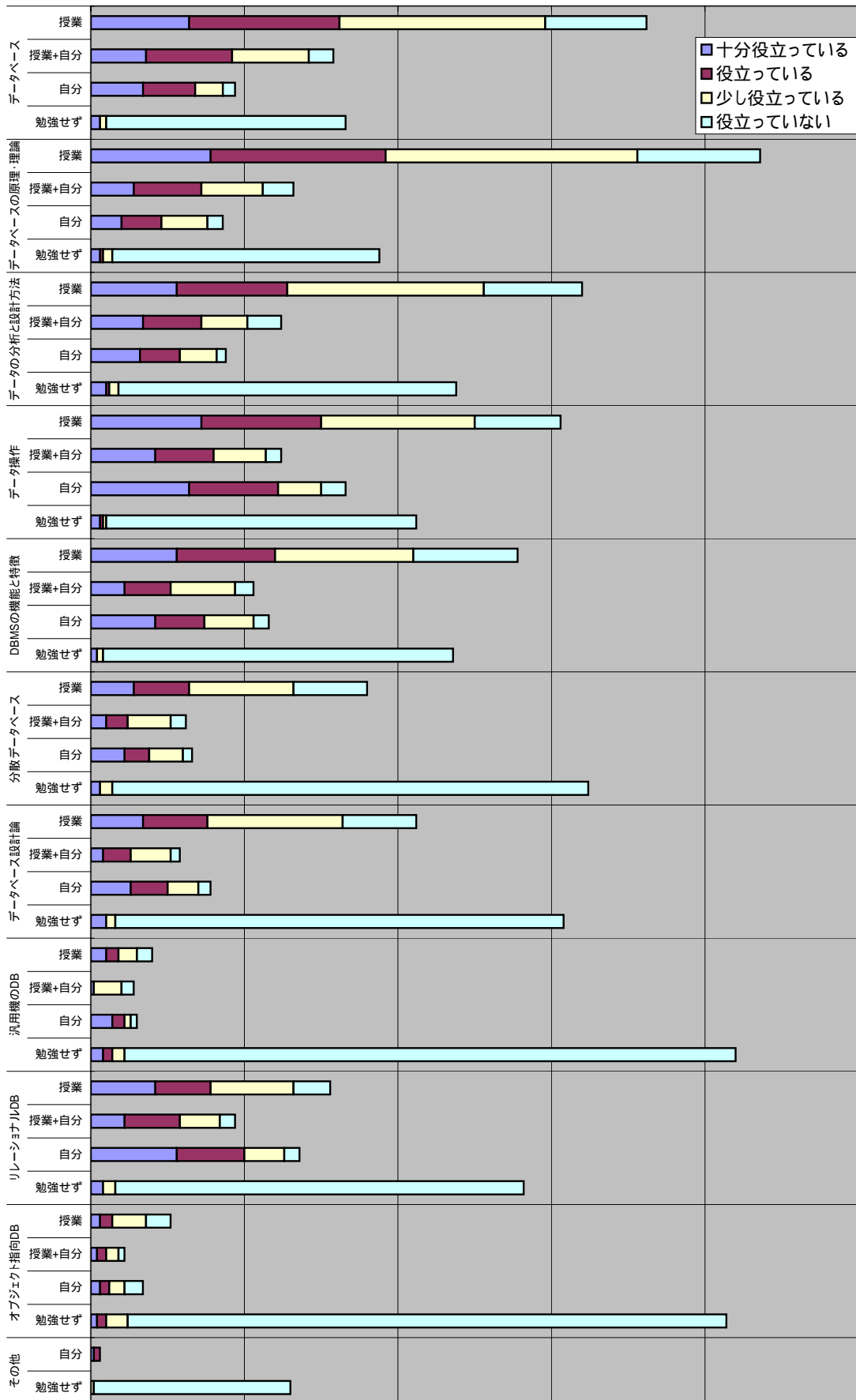


## 2 - 1 - c - 8 . 勉強方法と効果:データベース

		十分役立っている	役立っている	少し役立っている	役立っていない
データベース	授業	32	49	67	33
	授業+自分	18	28	25	8
	自分	17	17	9	4
	勉強せず	3		2	78
データベースの原理・理論	授業	39	57	82	40
	授業+自分	14	22	20	10
	自分	10	13	15	5
	勉強せず	3	1	3	87
データの分析と設計方法	授業	28	36	64	32
	授業+自分	17	19	15	11
	自分	16	13	12	3
	勉強せず	5	1	3	110
データ操作	授業	36	39	50	28
	授業+自分	21	19	17	5
	自分	32	29	14	8
	勉強せず	3	1	1	101
DBMSの機能と特徴	授業	28	32	45	34
	授業+自分	11	15	21	6
	自分	21	16	16	5
	勉強せず	2		2	114
分散データベース	授業	14	18	34	24
	授業+自分	5	7	14	5
	自分	11	8	11	3
	勉強せず	3		4	155
データベース設計論	授業	17	21	44	24
	授業+自分	4	9	13	3
	自分	13	12	10	4
	勉強せず	5		3	146
汎用機のDB	授業	5	4	6	5
	授業+自分	1		9	4
	自分	7	4	2	2
	勉強せず	4	3	4	199
リレーショナルDB	授業	21	18	27	12
	授業+自分	11	18	13	5
	自分	28	22	13	5
	勉強せず	4		4	133
オブジェクト指向DB	授業	3	4	11	8
	授業+自分	2	3	4	2
	自分	3	3	5	6
	勉強せず	2	3	7	195
その他	自分	1	2		
	勉強せず			1	64

勉強方法と効果: データベース

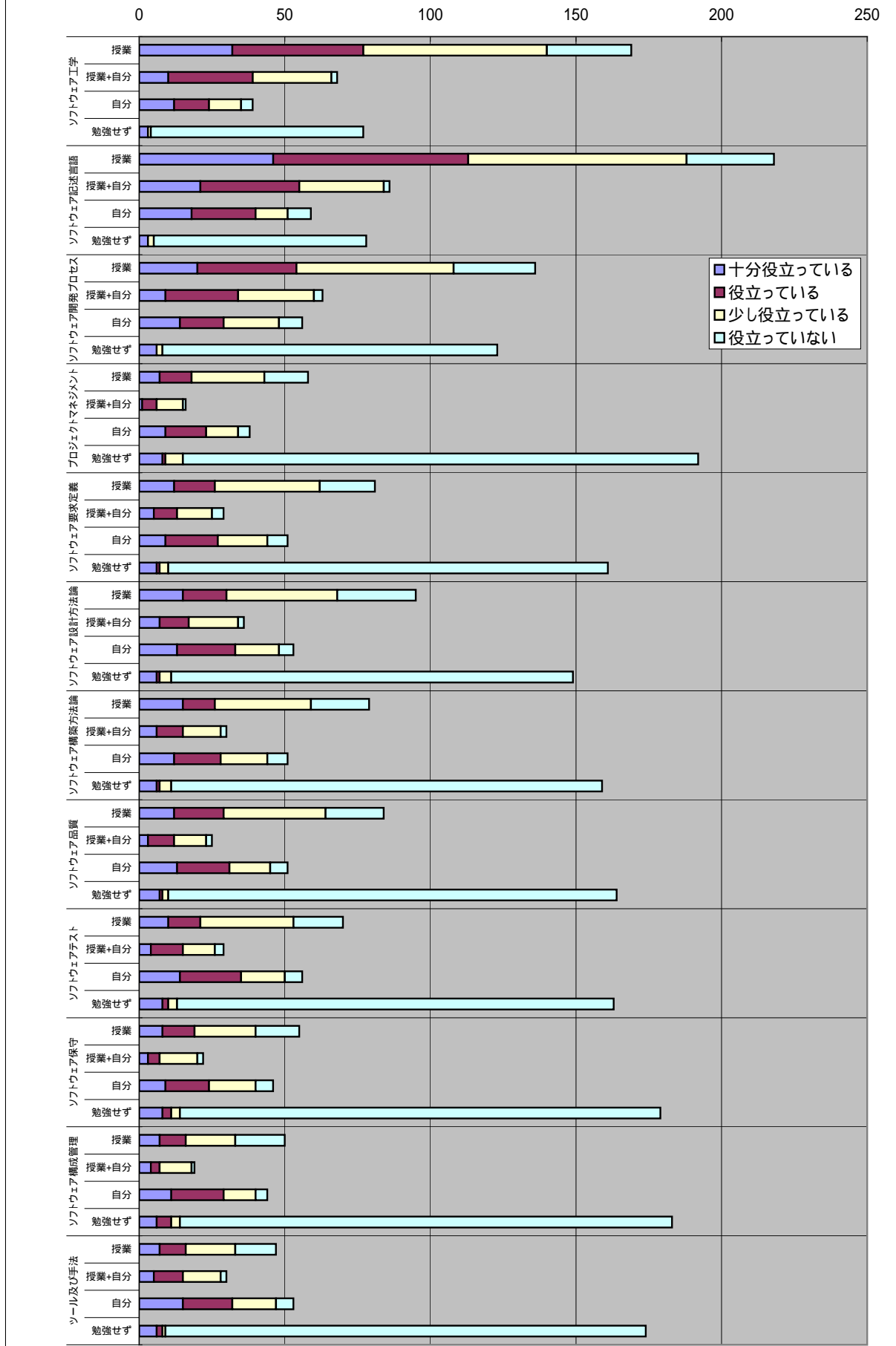
0 50 100 150 200 250



## 2 - 1 - c - 9 . 勉強方法と効果:ソフトウェア工学

		十分役立っている	役立っている	少し役立っている	役立っていない
ソフトウェア工学	授業	32	45	63	29
	授業+自分	10	29	27	2
	自分	12	12	11	4
	勉強せず	3		1	73
ソフトウェア記述言語	授業	46	67	75	30
	授業+自分	21	34	29	2
	自分	18	22	11	8
	勉強せず	3		2	73
ソフトウェア開発プロセス	授業	20	34	54	28
	授業+自分	9	25	26	3
	自分	14	15	19	8
	勉強せず	6		2	115
プロジェクトマネジメント	授業	7	11	25	15
	授業+自分	1	5	9	1
	自分	9	14	11	4
	勉強せず	8	1	6	177
ソフトウェア要求定義	授業	12	14	36	19
	授業+自分	5	8	12	4
	自分	9	18	17	7
	勉強せず	6	1	3	151
ソフトウェア設計方法論	授業	15	15	38	27
	授業+自分	7	10	17	2
	自分	13	20	15	5
	勉強せず	6	1	4	138
ソフトウェア構築方法論	授業	15	11	33	20
	授業+自分	6	9	13	2
	自分	12	16	16	7
	勉強せず	6	1	4	148
ソフトウェア品質	授業	12	17	35	20
	授業+自分	3	9	11	2
	自分	13	18	14	6
	勉強せず	7	1	2	154
ソフトウェアテスト	授業	10	11	32	17
	授業+自分	4	11	11	3
	自分	14	21	15	6
	勉強せず	8	2	3	150
ソフトウェア保守	授業	8	11	21	15
	授業+自分	3	4	13	2
	自分	9	15	16	6
	勉強せず	8	3	3	165
ソフトウェア構成管理	授業	7	9	17	17
	授業+自分	4	3	11	1
	自分	11	18	11	4
	勉強せず	6	5	3	169
ツール及び手法	授業	7	9	17	14
	授業+自分	5	10	13	2
	自分	15	17	15	6
	勉強せず	6	2	1	165

勉強方法と効果:ソフトウェア工学



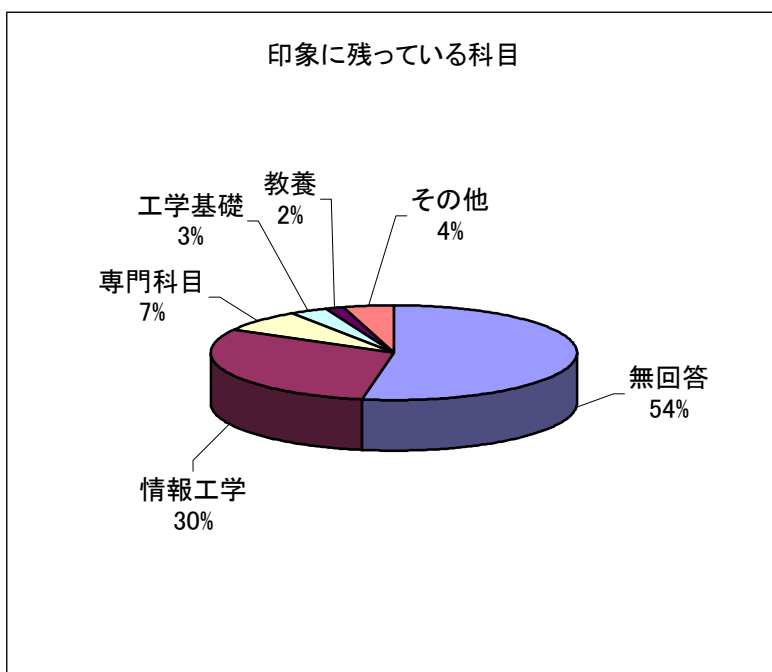
## 2-2. 印象に残っている授業について

2-2-a-1. 印象に残っている科目

2-2-a-2. 印象に残っている理由

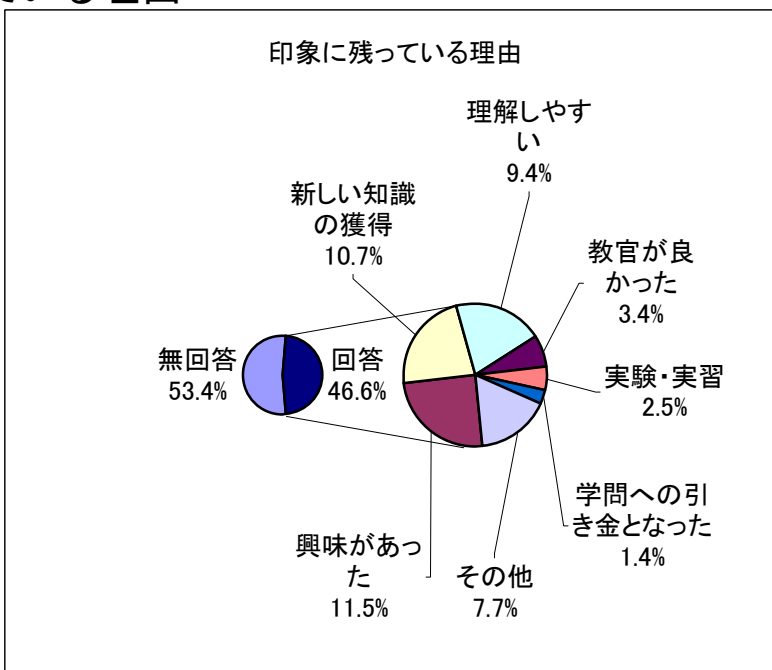
## 2-2-a-1. 印象に残っている科目

印象に残っている科目	件数
情報工学	196
専門科目	48
工学基礎	22
教養	10
その他	29
無回答	341
合計	646



## 2-2-a-2. 印象に残っている理由

印象に残っている理由	件数
興味があった	74
新しい知識の獲得	69
理解しやすい	61
教官が良かった	22
実験・実習	16
学問への引き金となった	9
その他	50
無回答	345
合計	646

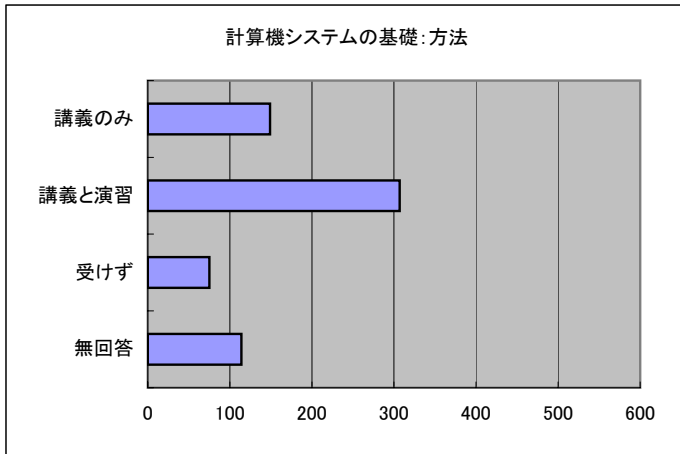


- 3. 教育方法について
- 3-1. 講義・演習について
- 3-1-1. 授業の方法と満足度について

- 3-1-1-a-1. 計算機システム基礎の勉強方法
- 3-1-1-b-1 : 計算機システム基礎の満足度
- ~
- 3-1-1-a-20 : その他のネットワーク系の勉強方法
- 3-1-1-b-20 : その他のネットワーク系の満足度

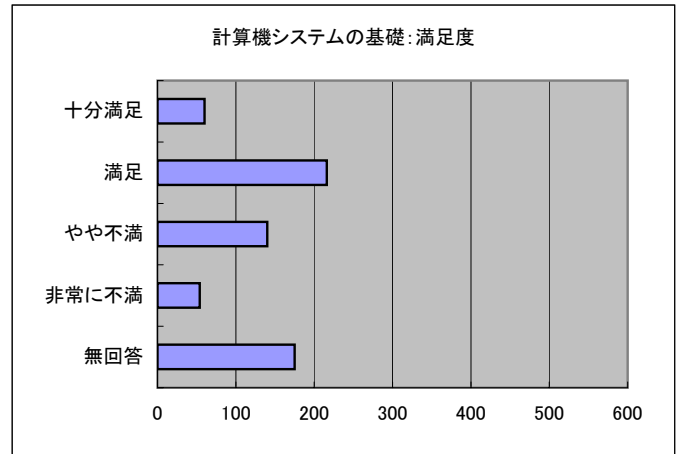
### 3-1-1-a-1: 計算機システム基礎の勉強方法

計算機システムの基礎:方法	人数
無回答	114
受けず	75
講義と演習	307
講義のみ	149



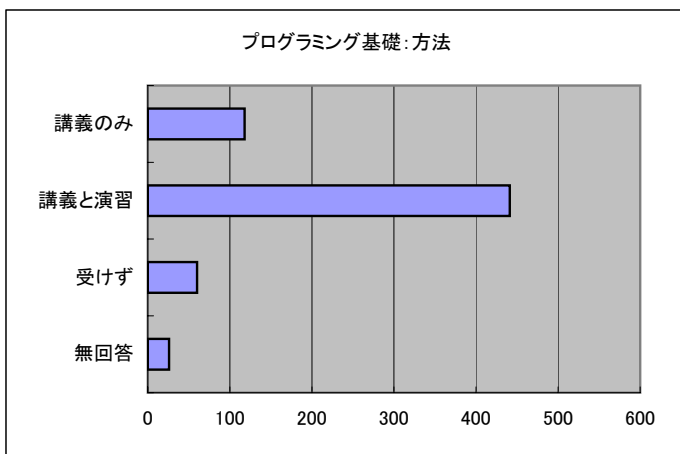
### 3-1-1-b-1: 計算機システム基礎の満足度

計算機システムの基礎:満足度	人数
無回答	175
非常に不満	54
やや不満	140
満足	216
十分満足	60



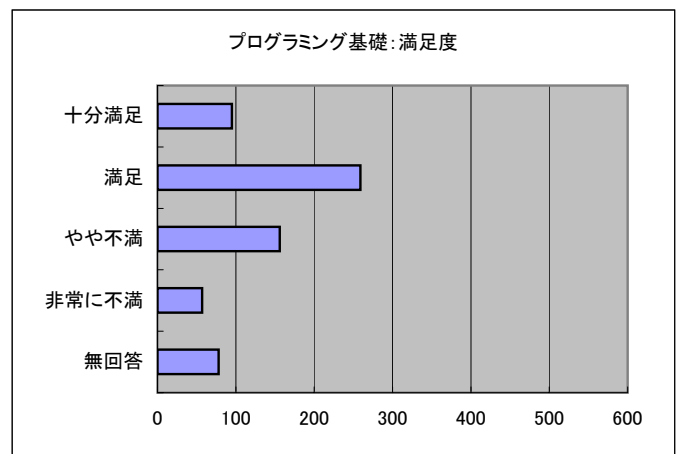
### 3-1-1-a-2: プログラミング基礎の勉強方法

プログラミング基礎:方法	人数
無回答	26
受けず	60
講義と演習	441
講義のみ	118



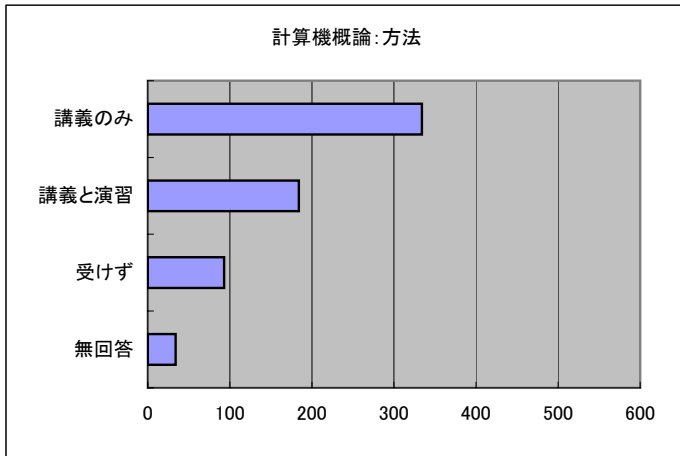
### 3-1-1-b-2: プログラミング基礎の満足度

プログラミング基礎:満足度	人数
無回答	78
非常に不満	57
やや不満	156
満足	259
十分満足	95



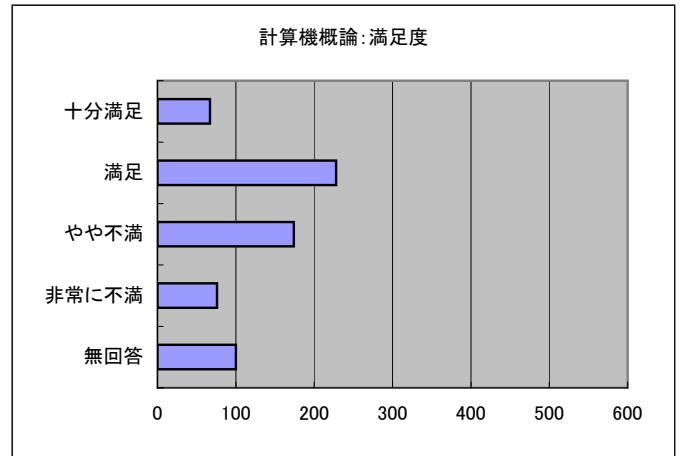
### 3-1-1-a-3: 計算機概論の勉強方法

計算機概論: 方法	人数
無回答	34
受けず	93
講義と演習	184
講義のみ	334



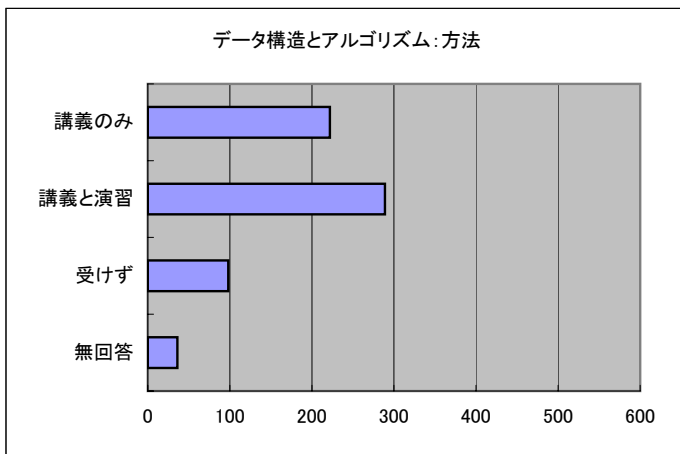
### 3-1-1-b-3: 計算機概論の満足度

計算機概論: 満足度	人数
無回答	100
非常に不満	76
やや不満	174
満足	228
十分満足	67



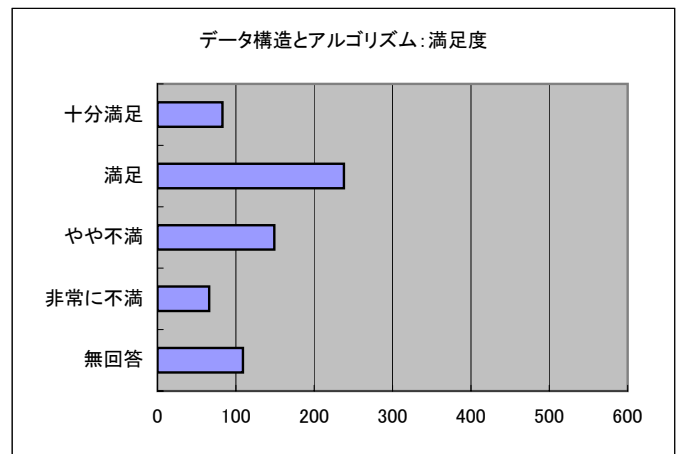
### 3-1-1-a-4: データ構造とアルゴリズムの勉強方法

データ構造とアルゴリズム: 方法	人数
無回答	36
受けず	98
講義と演習	289
講義のみ	222



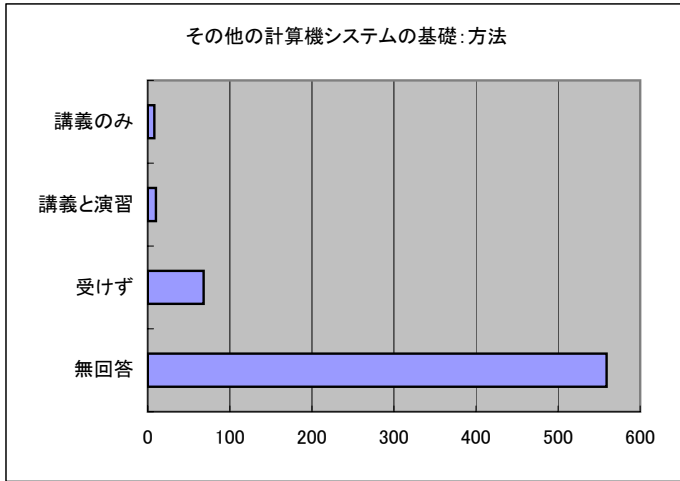
### 3-1-1-b-4: データ構造とアルゴリズムの満足度

データ構造とアルゴリズム: 満足度	人数
無回答	109
非常に不満	66
やや不満	149
満足	238
十分満足	83



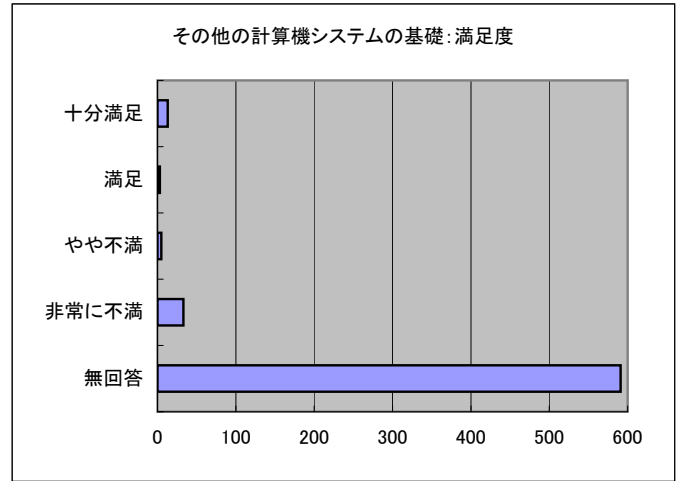
3-1-1-a-5: その他の計算機システムの基礎の勉強方法

その他の計算機システムの基礎:方法	人数
無回答	559
受けず	68
講義と演習	10
講義のみ	8



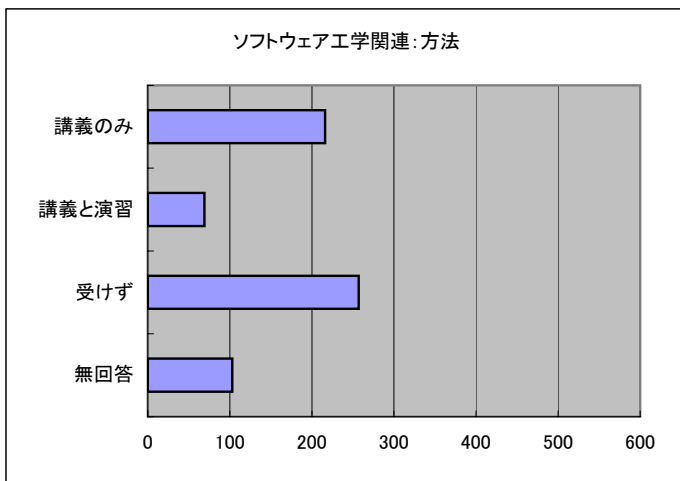
3-1-1-b-5: その他の計算機システムの基礎の満足度

その他の計算機システムの基礎:満足度	人数
無回答	591
非常に不満	33
やや不満	5
満足	3
十分満足	13



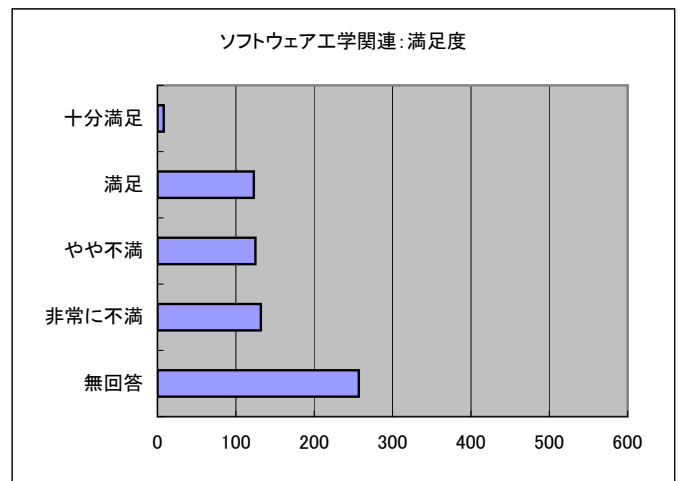
3-1-1-a-6: ソフトウェア工学関連の勉強方法

ソフトウェア工学関連:方法	人数
無回答	103
受けず	257
講義と演習	69
講義のみ	216



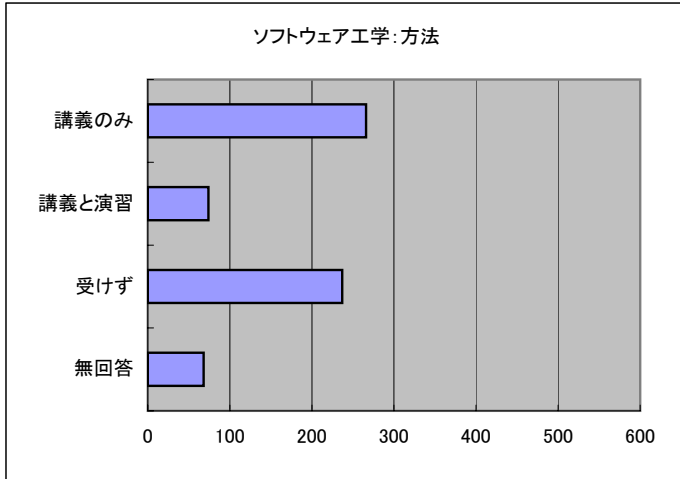
3-1-1-b-6: ソフトウェア工学関連の満足度

ソフトウェア工学関連:満足度	人数
無回答	257
非常に不満	132
やや不満	125
満足	123
十分満足	8



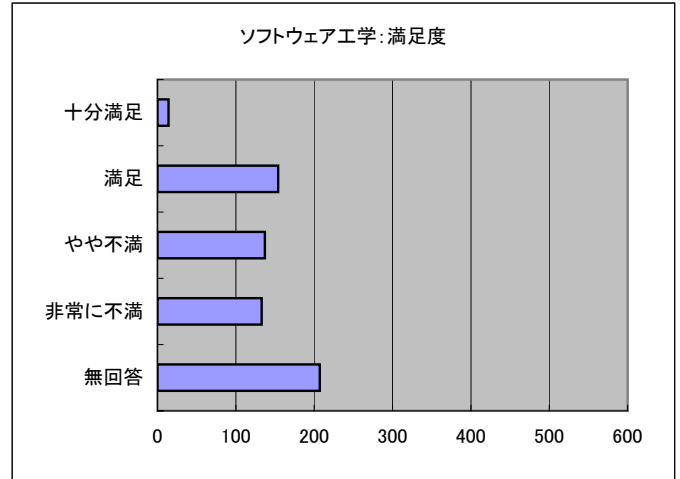
### 3-1-1-a-7:ソフトウェア工学の勉強方法

ソフトウェア工学:方法	人数
無回答	68
受けず	237
講義と演習	74
講義のみ	266



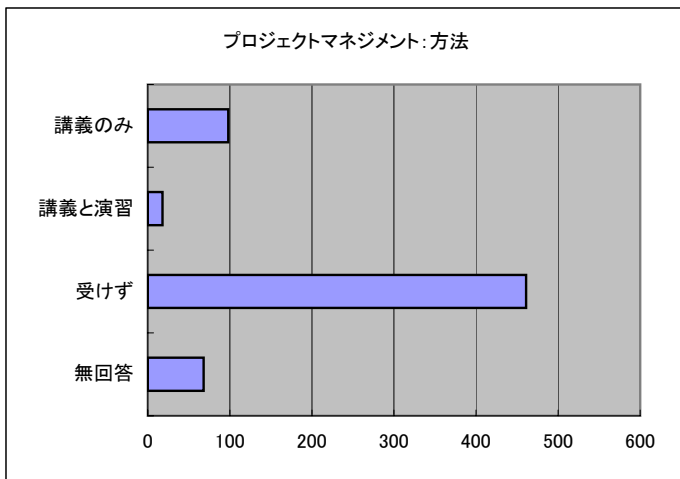
### 3-1-1-b-7:ソフトウェア工学の満足度

ソフトウェア工学:満足度	人数
無回答	207
非常に不満	133
やや不満	137
満足	154
十分満足	14



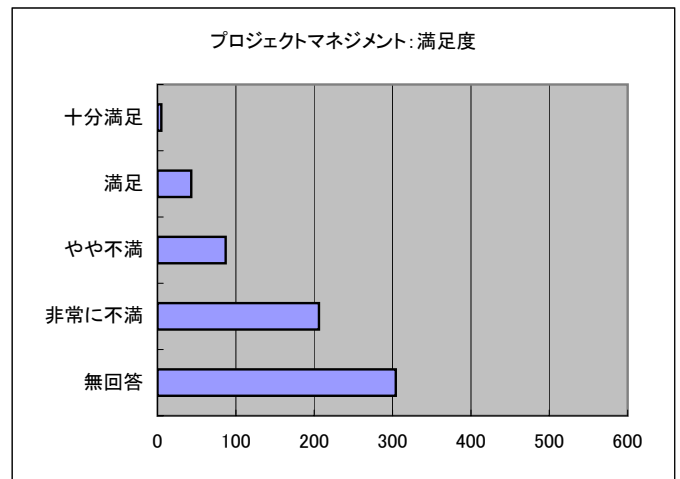
### 3-1-1-a-8:プロジェクトマネジメントの勉強方法

プロジェクトマネジメント:方法	人数
無回答	68
受けず	461
講義と演習	18
講義のみ	98



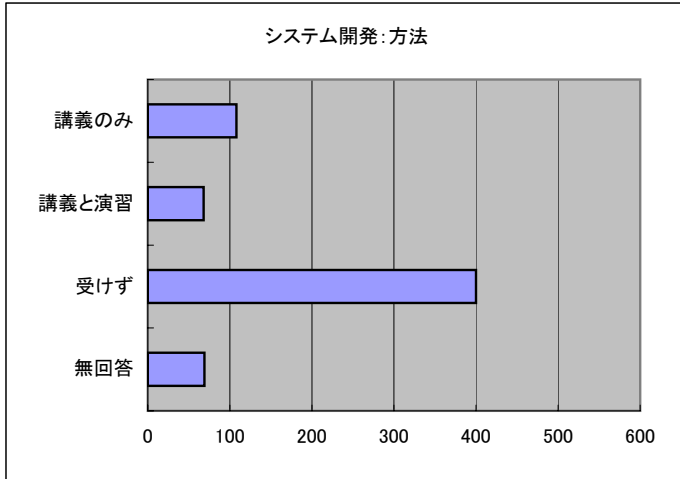
### 3-1-1-b-8:プロジェクトマネジメントの満足度

プロジェクトマネジメント:満足度	人数
無回答	304
非常に不満	206
やや不満	87
満足	43
十分満足	5



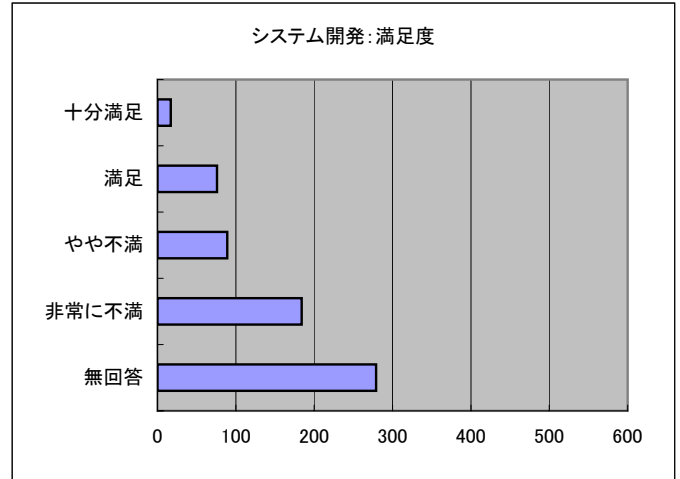
3-1-1-a-9:システム開発の勉強方法

システム開発:方法	人数
無回答	69
受けず	400
講義と演習	68
講義のみ	108



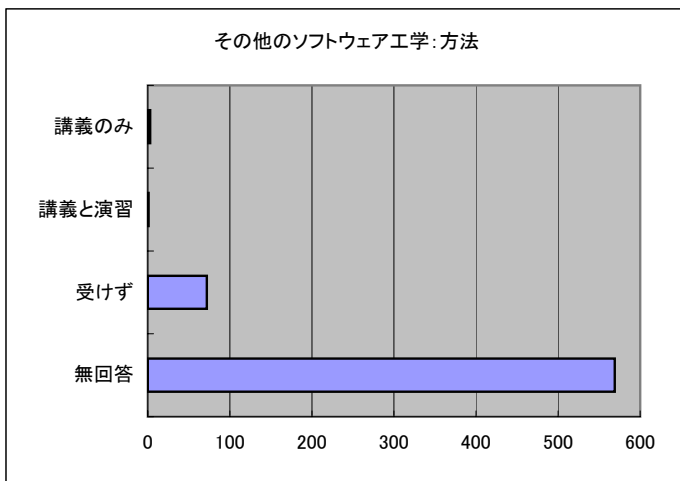
3-1-1-b-9:システム開発の満足度

システム開発:満足度	人数
無回答	279
非常に不満	184
やや不満	89
満足	76
十分満足	17



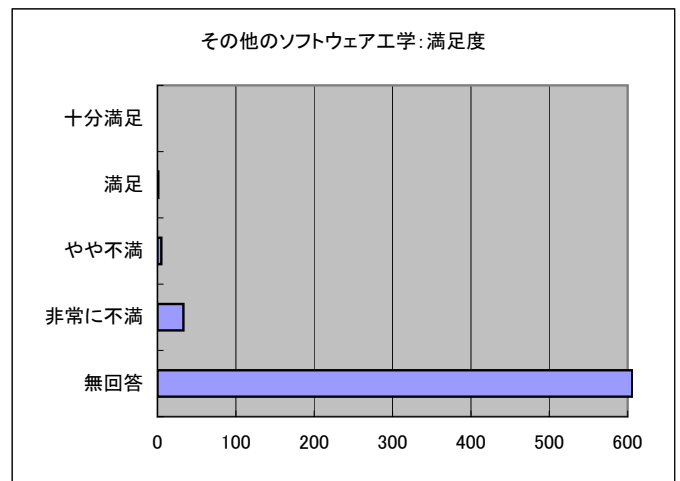
3-1-1-a-10:その他のソフトウェア工学の勉強方法

その他のソフトウェア工学:方法	人数
無回答	569
受けず	72
講義と演習	1
講義のみ	3



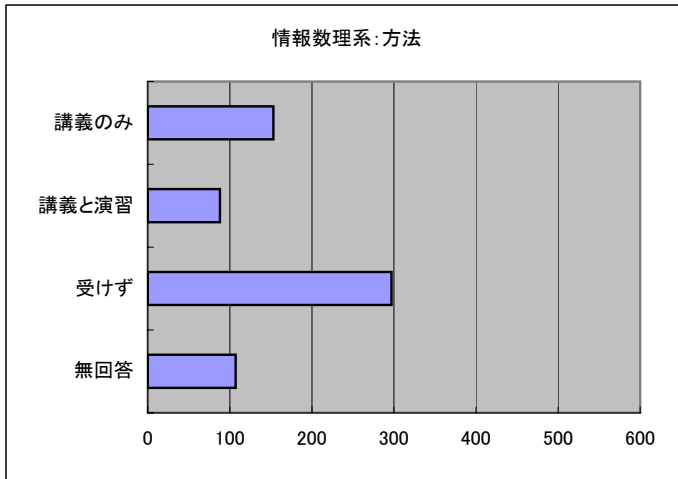
3-1-1-b-10:その他のソフトウェア工学の満足度

その他のソフトウェア工学:満足度	人数
無回答	606
非常に不満	33
やや不満	5
満足	1
十分満足	0



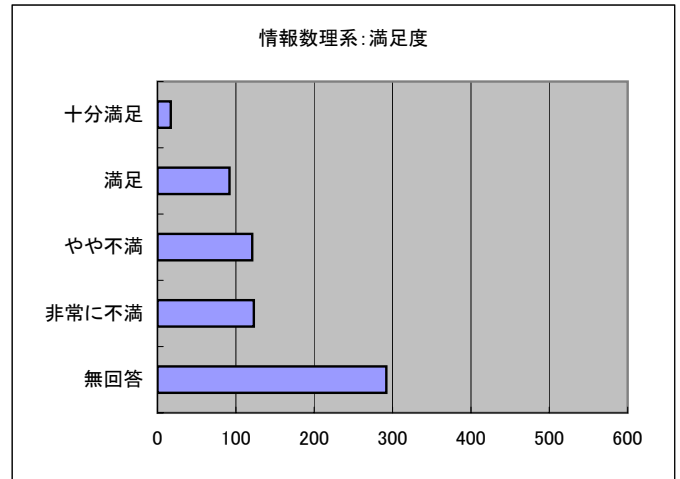
### 3-1-1-a-11:情報数理系の勉強方法

情報数理系:方法	人数
無回答	107
受けず	297
講義と演習	88
講義のみ	153



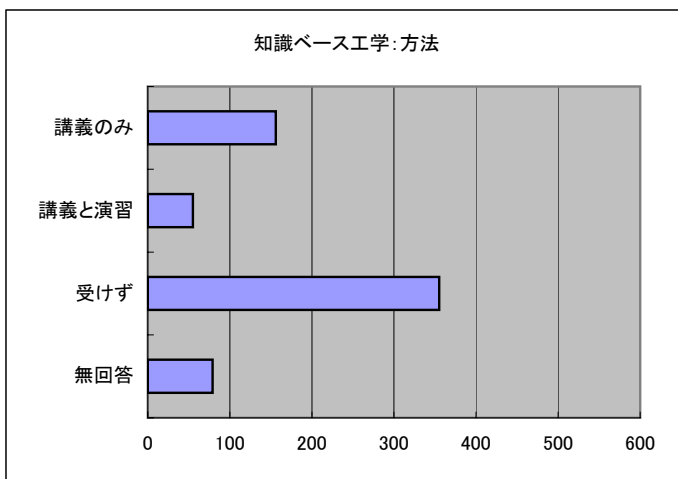
### 3-1-1-b-11:情報数理系の満足度

情報数理系:満足度	人数
無回答	292
非常に不満	123
やや不満	121
満足	92
十分満足	17



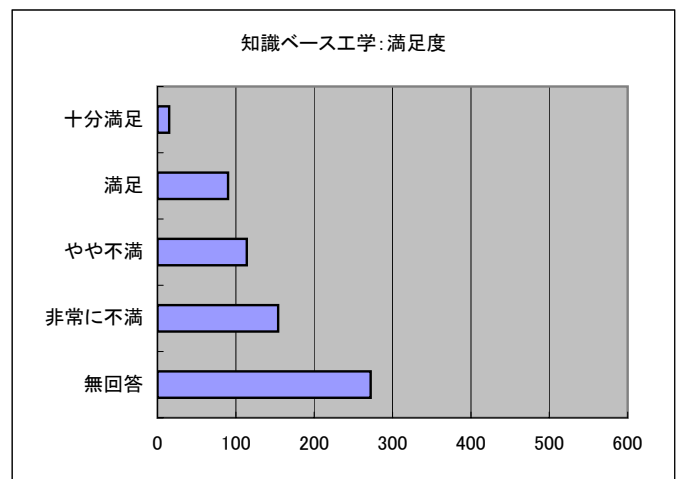
### 3-1-1-a-12:知識ベース工学の勉強方法

知識ベース工学:方法	人数
無回答	79
受けず	355
講義と演習	55
講義のみ	156



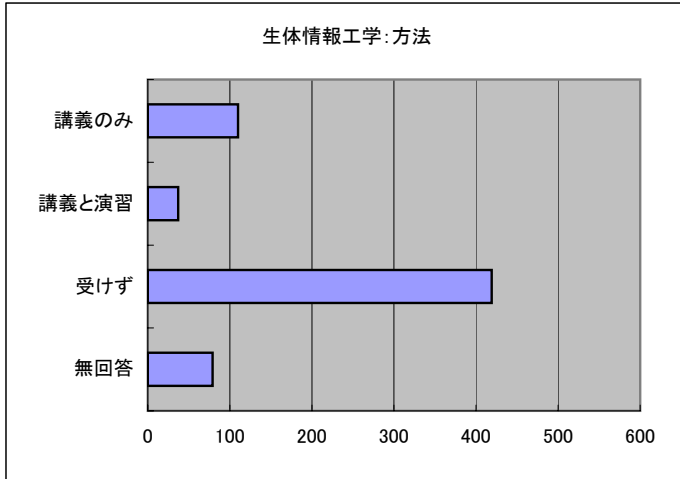
### 3-1-1-b-12:知識ベース工学の満足度

知識ベース工学:満足度	人数
無回答	272
非常に不満	154
やや不満	114
満足	90
十分満足	15



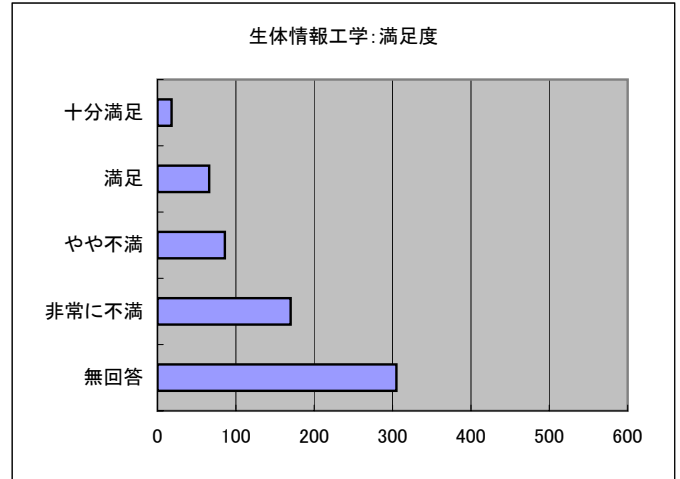
### 3-1-1-a-13: 生体情報工学の勉強方法

生体情報工学:方法	人数
無回答	79
受けず	419
講義と演習	37
講義のみ	110



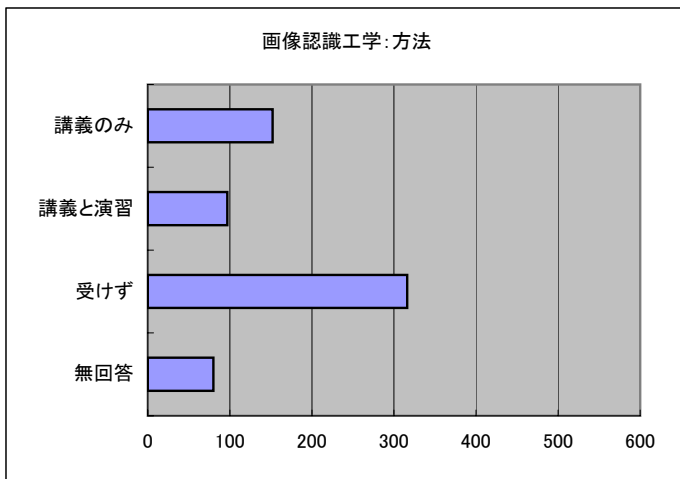
### 3-1-1-b-13: 生体情報工学の満足度

生体情報工学:満足度	人数
無回答	305
非常に不満	170
やや不満	86
満足	66
十分満足	18



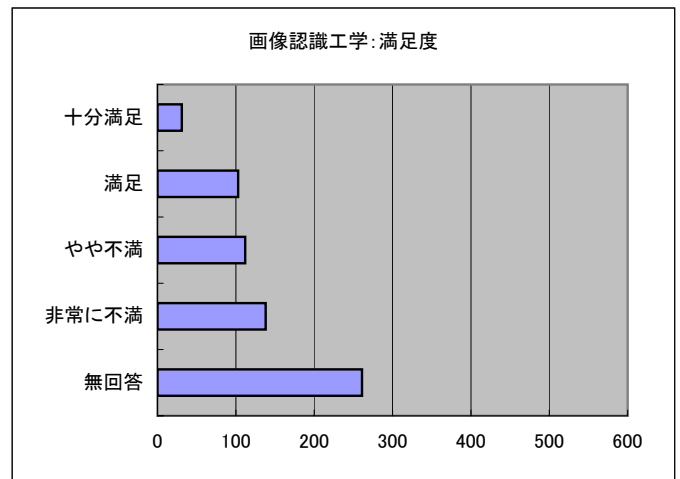
### 3-1-1-a-14: 画像認識工学の勉強方法

画像認識工学:方法	人数
無回答	80
受けず	316
講義と演習	97
講義のみ	152



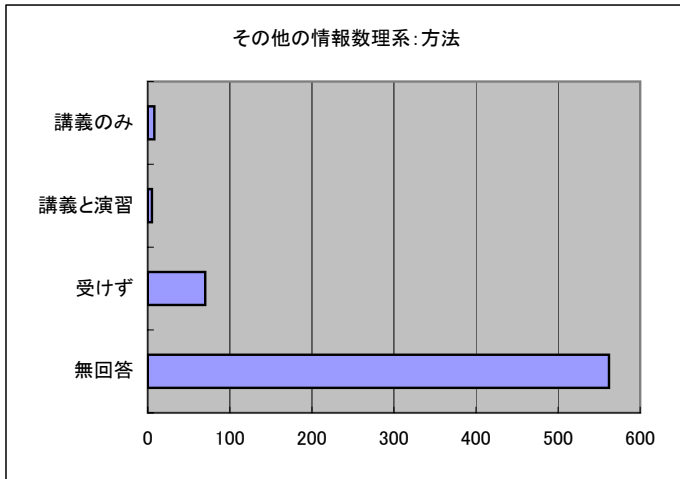
### 3-1-1-b-14: 画像認識工学の満足度

画像認識工学:満足度	人数
無回答	261
非常に不満	138
やや不満	112
満足	103
十分満足	31



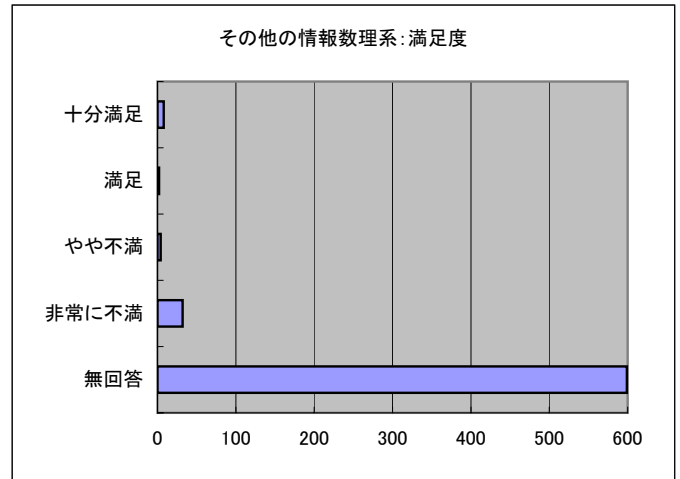
3-1-1-a-15: その他の情報数理の勉強方法

その他の情報数理系:方法	人数
無回答	562
受けず	70
講義と演習	5
講義のみ	8



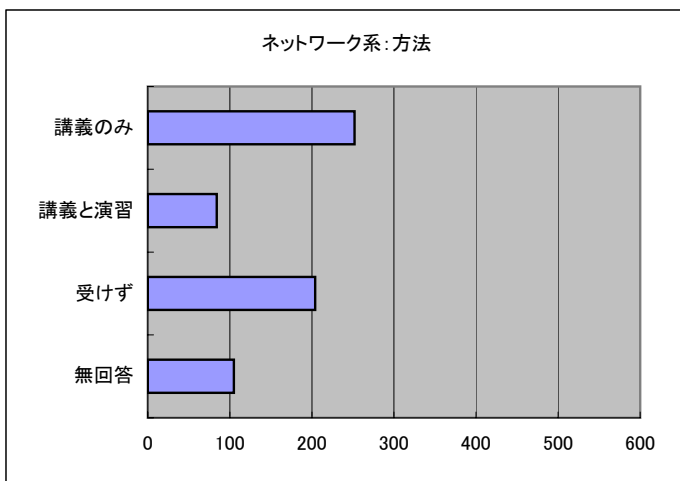
3-1-1-b-15: その他の情報数理の満足度

その他の情報数理系:満足度	人数
無回答	599
非常に不満	32
やや不満	4
満足	2
十分満足	8



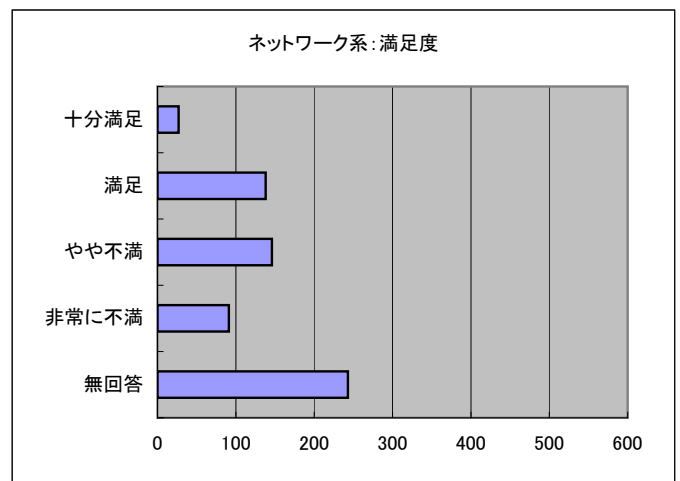
3-1-1-a-16: ネットワーク系の勉強方法

ネットワーク系:方法	人数
無回答	105
受けず	204
講義と演習	84
講義のみ	252



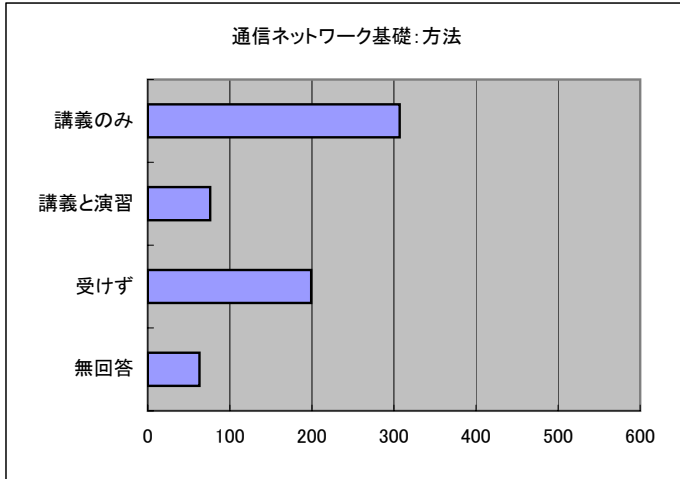
3-1-1-b-16: ネットワーク系の満足度

ネットワーク系:満足度	人数
無回答	243
非常に不満	91
やや不満	146
満足	138
十分満足	27



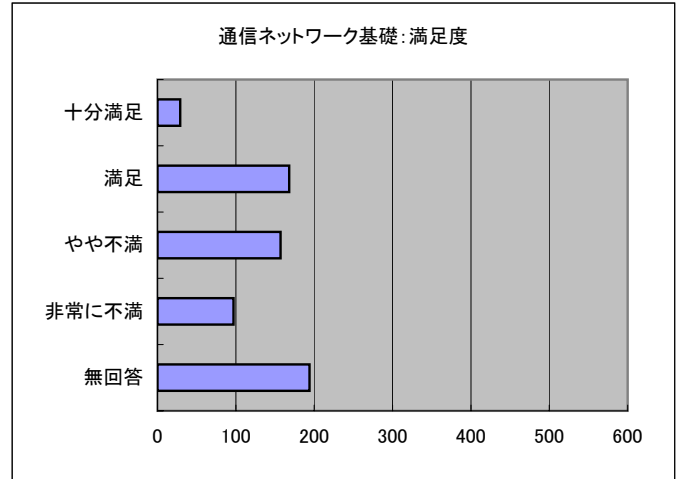
3-1-1-a-17:通信ネットワーク基礎の勉強方法

通信ネットワーク基礎:方法	人数
無回答	63
受けず	199
講義と演習	76
講義のみ	307



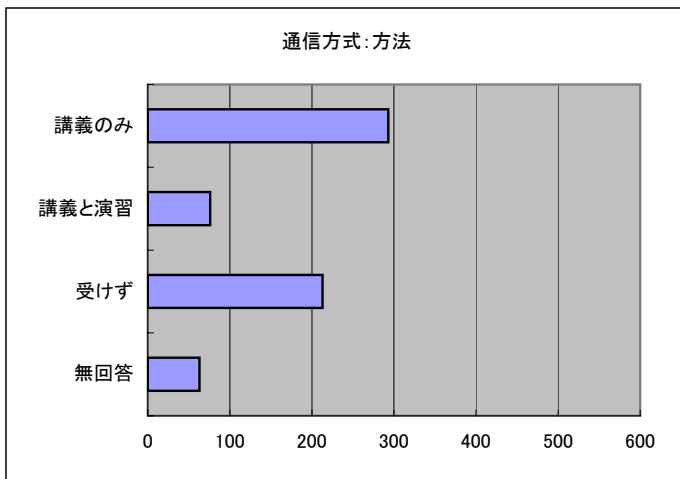
3-1-1-b-17:通信ネットワーク基礎の満足度

通信ネットワーク基礎:満足度	人数
無回答	194
非常に不満	97
やや不満	157
満足	168
十分満足	29



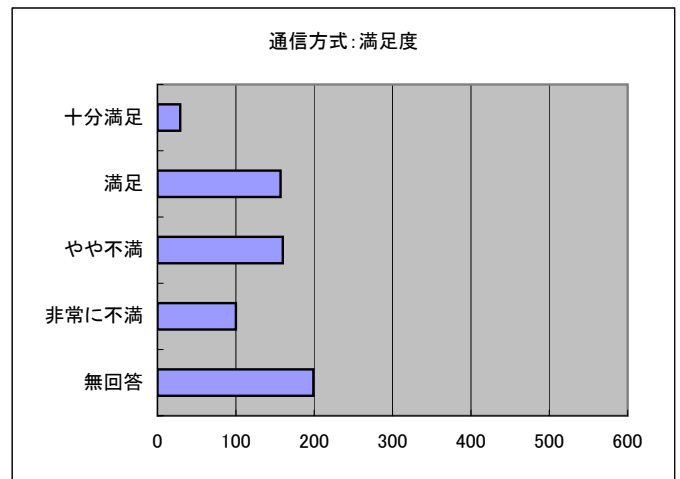
3-1-1-a-18:通信方式の勉強方法

通信方式:方法	人数
無回答	63
受けず	213
講義と演習	76
講義のみ	293



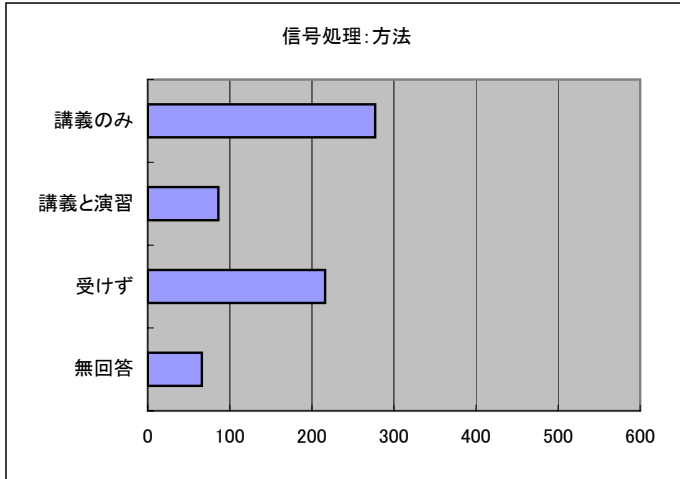
3-1-1-b-18:通信方式の満足度

通信方式:満足度	人数
無回答	199
非常に不満	100
やや不満	160
満足	157
十分満足	29



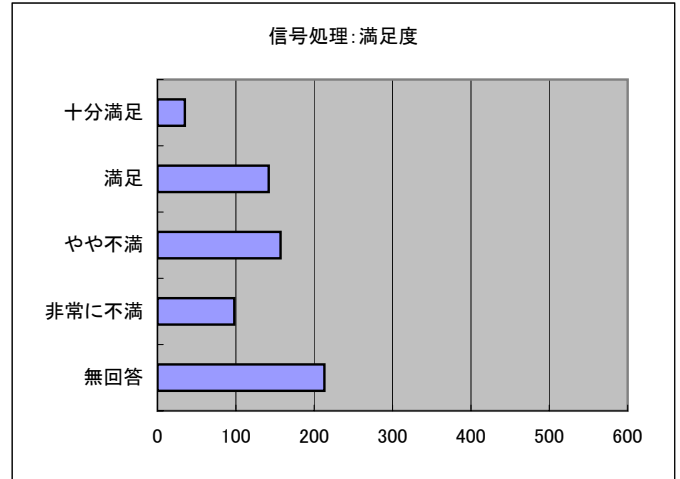
3-1-1-a-19: 信号処理の勉強方法

信号処理:方法	人数
無回答	66
受けず	216
講義と演習	86
講義のみ	277



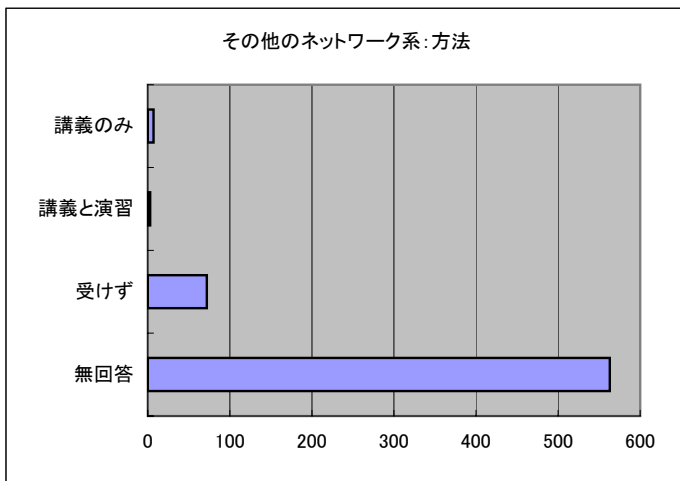
3-1-1-b-19: 信号処理の満足度

信号処理:満足度	人数
無回答	213
非常に不満	98
やや不満	157
満足	142
十分満足	35



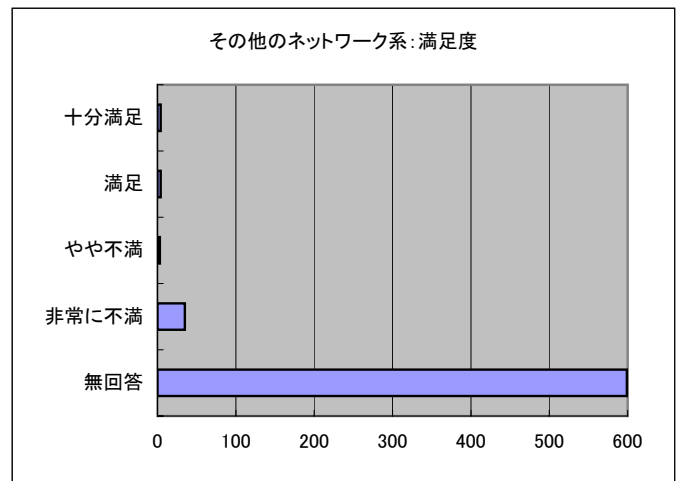
3-1-1-a-20: その他のネットワーク系の勉強方法

その他のネットワーク系:方法	人数
無回答	563
受けず	72
講義と演習	3
講義のみ	7



3-1-1-b-20: その他のネットワーク系の満足度

その他のネットワーク系:満足度	人数
無回答	599
非常に不満	35
やや不満	3
満足	4
十分満足	4



### 3-1-2. 授業の過不足について

- 3-1-2-a-1. 授業の過不足： 計算機システムの基礎
- 3-1-2-a-2. 授業の過不足： ソフトウェア工学関連
- 3-1-2-a-3. 授業の過不足： 情報数理系
- 3-1-2-a-4. 授業の過不足： ネットワーク系

### 3-1-2-a-1. 授業の過不足: 計算機システムの基礎 (人数)

計算機システムの基礎	追加	実習/演習/応用事例	62
		コンピュータアーキテクチャ	33
		オブジェクト指向言語	15
		データ探索/アルゴリズム	8
		情報的な考え方(基礎知識)	7
		パソコンのハード	7
		その他	26
	特にありませんが、今現在教えている内容が、将来にどういった影響を与えるかということを、授業の中で伝えてほしい。どのように役立つかとか。		
	不要	プログラミング言語	9
		FORTRAN,Pascal,C	
電気・電子回路		5	
オートマトン		4	
その他		26	
人工知能、計算機プログラミング(アセンブル) コンピュータの歴史			

### 3-1-2-a-2. 授業の過不足: ソフトウェア工学関連 (人数)

ソフトウェア工学関連	追加	ソフトウェア工学	78
		システム開発演習	63
		プロジェクトマネジメント	44
		プログラミング	16
		データベース	6
		その他	22
	不要	プログラミング言語	12
		Javaやオブジェクト指向等は講義だけでは理解できない。理解できる人は講義に出てない。 COBOL, FORTRAN, Pascal	
		その他	12
		大学教育で一般的な応用を教えること自体不要。基礎技術の習得と歴史的に優れた応用例を少し紹介するだけであとは考えさせる。	

### 3-1-2-a-3. 授業の過不足: 情報数理系 (人数)

情報数理系	追加	演習・実習	20
		画像処理	12
		数学・理論	10
		データマイニング	6
		人工知能関連	7
		その他	34
		経営工学系科目、例えばゲーム理論	
	不要	ファジー/カオス/人口知能	9
		情報数学	4
		オートマトン	2
その他	17		
生体情報工学問いう名の電極やオペアンプの話			

### 3-1-2-a-4. 授業の過不足: ネットワーク系 (人数)

ネットワーク系	追加	ネットワーク構築演習	72
		ネットワーク基礎	61
		プロトコル/標準	30
		セキュリティ	23
		その他	33
	不要	TCP/IP以外のプロトコル	4
		その他	11
		ATM/STM、電話交換機の話 応用科目(基礎ができていなければ意味なし)	

## 3-2. PBL、インターンシップについて

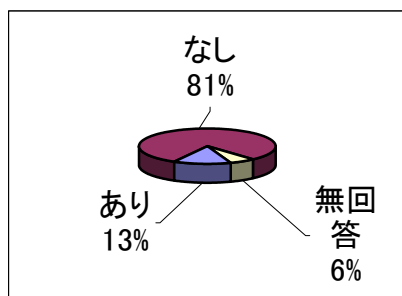
### 3-2-1. 経験について

- 3-2-1-a-1. PBLの経験の有無
- 3-2-1-b-1. インターンシップの経験の有無
- 3-2-1-b-2. 一回目のインターンシップの期間
- 3-2-1-b-3. 二回目のインターンシップの期間
  
- 3-2-1-c-1. PBLのテーマ
- 3-2-1-c-2. PBLへのご意見
  
- 3-2-1-d-1. インターンシップのテーマ
- 3-2-1-d-2. インターンシップへのご意

※PBL:Project Based Learningの略。課題をもとにグループで実演習する学習方法。

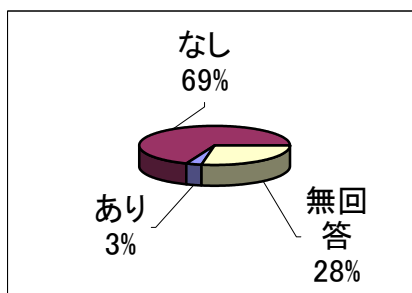
### 3-2-1-a-1. PBLの経験の有無

あり	87
なし	518
無回答	40



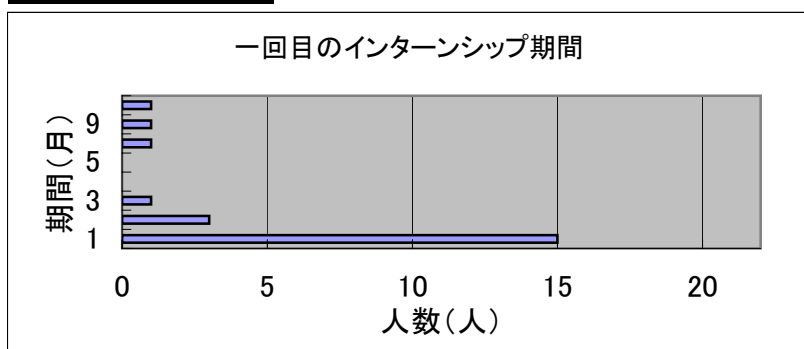
### 3-2-1-b-1. インターンシップの経験の有無

あり	22
なし	445
無回答	179



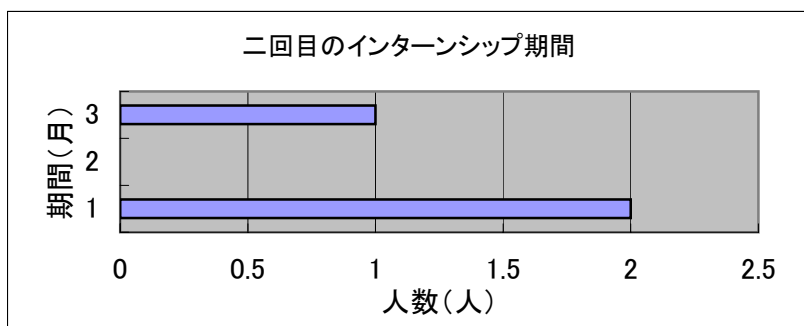
### 3-2-1-b-2. 一回目のインターンシップの期間

期間(月)	人数	平均: 2.3ヶ月
1	15	
2	3	
3	1	
4	0	
5	0	
6	1	
9	1	
12	1	



### 3-2-1-b-3. 二回目のインターンシップの期間

期間(月)	人数	平均: 1.7ヶ月
1	2	
2	0	
3	1	



### 3 - 2 - 1 - c - 1 . PBLのテーマ

課題名-分類	課題名(抜粋)	件数
設計・開発・構築	AIを用いたOne - 2 - one マーケティング支援システム	41
	Javaアプリケーションの作成	
	Javaによる、ATMシステムの作成	
	Mail Client Software の開発	
	PSゲームプログラミング	
	Windows上で動くゲームプログラム	
	アセンブラによる四則演算機開発	
	アセンブラの作成、実行	
	コンパイラ演習	
	システム構築演習	
	ショッピングWebサイトの構築	
	ソフトウェア開発	
	データベースシステムの開発演習	
	データベース設計・開発	
	ネットワーク通信、マルチタスク	
	プログラミング演習	
	プロセッサ設計	
	マイクロプログラムの設計	
	ロボカップ	
	研究室で使用するオフィススケジューラの作成	
	工学設計	
	自律移動ロボットの製作	
	情報工学実験	
	情報実験	
	情報処理	
	探索アルゴリズムを用いたソフト開発	
和英翻訳プログラムの作成		
その他情報系	PICを用いたマイコン作成	20
	アプレットを利用したアニメーションの作成	
	カオス理論に基づくふりこの動き(?)	
	システム分析演習	
	ソフトウェア実験	
	ソリューションビジネス発案	
	データベース設計	
	ハードウェア演習	
	ロボットの動作実験	
	管理工学実験	
	経営工学実験	
	自然科学実験	
情報系学問	コンパイラシステムの構築	8
	コンパイラ設計	
	データ解析	
	画像処理	
	計算機システムのしくみ	
通信工学実験		

課題名-分類	課題名(抜粋)	件数
情報技術	IE(上の設問で回答) Javaによる3Dアニメーション Javaの言語を使ったプログラム入門 Windows2000について エレベータ制御 ソフトウェア実験 プログラム演習 ロボットアーム制御 実験	9
その他	HP作成 サーマットボードによるフリップフロップ作成 ハードウェア実験 ビデオ会議システムのセールス資料作成 英米演劇 演習(3年) 遠隔医療 基礎実験 建築構造力学(スパゲティ演習) 工学設計 社会工学演習 修学計画 小説紹介 地図作成 電気工学演習 電気実験演習 電子回路? 特になし(実験) 品質管理 物理学実験	20

### 3-2-1-c-2. PBLへのご意見

教育効果-分類	教育効果
<p>良い</p>	<p>1つの仕事を協同してやってゆくという作業の基本が学べて有効だと思う</p> <p>Windows1に対する認識が単なるソフトウェアからOSとなった。ネットワークの導入(入門)を部分で、TCP/IPなどを学習するよりも、ワークグループで実際に小さいLANを構築した方がネットワークへの興味が湧いた。また、組織の一員として自分のふるまい方、問題に対して、グループで解決する方法を、疑似体験的に学ぶことで習得できたように感じる。</p> <p>グループで作業をすることによるメリット・弊害に気が付きながら、各メンバーが自分の役割を自覚し、どのようにグループ作業に貢献していくことができるかを考えることは有用である。</p> <p>グループ単位で動き、グループで課題をクリアする。実務では非常に重要な事だと思います。</p> <p>グループ内での議論や課題克服など有意義だと思う。</p> <p>コンパイラの仕組みや、システム開発の方法を学ぶ上では非常に役に立つ内容だった。</p> <p>しっかりとした手今があり、それについて日々勉強することができるため毎週行われる講義と比べると、得た知識は数倍であったと思う。</p> <p>チーム内での意思統制、進捗管理などのプロジェクトマネジメント面と、課題解決を行う上で必要な知識、スキルを問われるこのPBLは正直面白い。ある程度の期間があれば、知識ゼロの状態でも始めても良い。</p> <p>ドキュメント作成等基本的なことから、お客様との折衝・メンバーとの連携の重要性を理解できる有効な研修でした。</p> <p>やる気さえあればレベルの高いシステムを作るために自ら勉強をしたくなるのでよいと思う</p> <p>一人だけのプログラムとは違い、全体を運営する大変さや、作業の割り振りなど？との協力の難しさを学ぶいい機会だったと思う。</p> <p>会社ではグループ作業の占める割合が多いため、非常に役に立った。</p> <p>楽しく学習することができる。</p> <p>記憶に残る授業で、楽しかった。また、発表したり、グループ内で議論したり、グループで競争したりと総合的な学習が出来たと思う。</p> <p>作業の切り分け、割り振りなど、一人では考える必要のない部分であるが、社会に出るからは必要なものであり、よい練習になったと思う。</p> <p>実際にグループで作業することの難しさが分かる。とても良い。</p> <p>小さな規模だが、チーム単位での活動のよい練習になったと思う。</p> <p>他の人の意見や考え方を聞く良い機会であると思います。また、協力する事でチームでの進め方や行動の仕方が学べますし、一人一人に責任感がうまれる為、身になりやすいと思います。</p> <p>単に一人で机やPCに向かうよりも、コミュニケーションがあり、楽しさが増す。一人よりもアイデアが浮かぶ。反面、なれ合いが起こることも多々あるので、上は上、下は下のグループ分けが必要。</p> <p>仲間とやるのは、結構良いかもしれません。</p> <p>当初はクイズ感覚であたってみていたが、本当に会社に入ってみて様々な問題に直面するので、このような業務上の問題を例にとり考えることは、将来的に役に立つと思う。</p> <p>考える力が身につくようになったと思う。</p> <p>討論することで、研究範囲が広がり、より多くのパターンを実験でき、結果良い物ができ上がったと思う。深くとり組むことができる。</p> <p>非常に役立っています。</p> <p>本学では「工学設計」教育があり、1つのテーマに対し、グループで話し合いを重ね、1つの提案を行う。プレゼン技法、ポスターセッションの技法等、大変楽しく良かったと思う。</p> <p>理論的な内容をPBLによってイメージすることによって、理解が深まった。とても効果的である。</p> <p>話し合い、自分達で調べることは、とても勉強になった。</p>

教育効果-分類	教育効果
何かが身につく	<p>(大学入学してすぐに行うより、効果があるかもしれない)作業報告、メンバーシップの大切さが分かった。</p> <p>PBLの講義内容を学ぶだけでなく、複数人数での作業を期限までに完了させるという目標をグループで協力しながら実現することによって、時間・作業量・品質などのシステム開発に必要な要素を自然と身につけていくことができるから。</p> <p>グループで協力しながら目標を達成することは、単純に講義を受けるより大変だが身につく。</p> <p>グループで行動する中で、自分達の数日間のプロジェクトのマネジメントが学べる。(それがいかに難しいかということ)</p> <p>グループワークにより、仕事の分担を行い、全員で一つの物を付く利上げる経験ができる。</p> <p>グループ演習で役割分担や意思疎通、進捗管理助け合いなど、共同作業の基本を身に付けることができる。問題点の解決を皆で行った経験は貴重です。</p> <p>グループ及び意見を交換することで1人で創るよりも高い技術のモノを作成することができる。</p> <p>グループ作業のマネジメントの難しさが分かれば充分である。この時に、リーダを進んで経験することが、後々きいてくると思う。</p> <p>ソフトウェアの設計で、自分の作りたいシステムの提案などから、自分の意見等をグループメンバーや教師にどう提案していけば良いかを得られる。</p> <p>ソフトウェアの分散開発が体験できた。仕様の討議や開発、結合テスト、ドキュメントの作成課程で生じる諸問題と解決方法の模索が体験できた。</p> <p>みんなで考えることで、クオリティーの統一の難しさや、アプリケーションを作成する手順を勉強できた</p> <p>課題にもよりますが、組織活動訓練の効果は大きいと思われまます。</p> <p>協力し合うことの大切さ、人によって感じ方が違うことを学べた。</p> <p>仕様決定の重要性を認識できる。</p> <p>集団で議論し、方式をつめていくなどの点で、実社会における仕事の仕方に通じるものが多い。一方で、形式化した点があり、手抜き等をしようと思えばいくらでも出来てしまう点で、課題の意図が掴めない者には意味がない。</p> <p>人それぞれの疑問点は違うので、新たな発見が多いと思う。</p> <p>同じ目的(要求)のもとに、同じシステムの設計を行っても様々な手法や考え方があることが認識でき、実演習を通して大まかではあるが、一連の流れを理解できる。</p> <p>必要な科目であると思う。人と話す訓練になる。しかし、真剣に行わないと損する。複数人での仕事は、それぞれの進捗異言や情報の共有が重要であることを学びました</p>
悪い/問題点がある	<p>グループでの取り組みということでも意義なものであったが、できる人に全てまかせるといった傾向も見られ、プロジェクトのまとめ方にも問題があるように感じた。</p> <p>一人一人の役割等が、いまいちハッキリせず、出来る人がやって、出来ない人は分からないままだった。</p> <p>一部の人がメインで行ってしまう</p> <p>開発言語(C)の習熟度は向上するが、実際のプログラム開発方法とはかけ離れている。このあたりの演習も必要なのではないか。</p> <p>指導教官の指導内容により、その教育効果は非常にまちまちであると思う。それぞれの開発フェーズ毎に適切に目標・目的を設定してくれた教官もいたが、一部、放任している教官もいたため、最終的には成果物に大きな差が見られた。</p> <p>内容を「体感」できる点では有効だが、必ずしも企業レベルの訓練とはならないだろう。(指導する助手やアシスタントも実地経験を持っていないため)</p> <p>グループで、作業スケジュールを決定するため、個々の役割が不明確であった。出来る人、知っている人に任せていたという思い出がある。</p> <p>グループ内で一人でやってしまう人がいて、問題意識・解決策の共有がでなかった。</p> <p>グループ内のメンバーに知識的なかたよりがあると、そのメンバーの負荷が高くなってしまう場合がある。</p> <p>レベルがバラバラでは、あまり効果的ではない。</p> <p>人まかせになるのであまりよい気がしない</p> <p>ICをクリーンルームで焼いたり、ふだんできないようなことの方が印象深い。情報系でも単にプログラムを書く、システムを組む、だけでは面白くないし、わからないのでは。</p>
その他	<p>落ちこぼれずに済んだ</p> <p>もっと有効活用してほしい</p>

### 3 - 2 - 1 - d - 1 . インターンシップのテーマ

テーマ-分類	テーマ	件数
ソフト開発	BASICで書かれたソースコードの自動変換	17
	GIS	
	Javaによる教育コンテンツ作成	
	Javaを使つての携帯アプリケーション	
	Java言語の習熟と演習	
	MD DATA ドライバの作成	
	Virtual Reality におけるマシンインターフェース用ソフトの開発	
	Webシステムの改善	
	オンロジニサーバのクライアント開発	
	オンラインシステムの開発	
	ソフトウェアエージェントの開発	
	ソフトウェア開発の工程を学ぶ	
	プログラミング支援ツールのプロトタイピング	
	ホームページの作成	
	モータードライバの開発	
	画像処理による不良品判別検査の自動化	
	顔認識に関する研究	
その他	ADSLのサービス提供	15
	CADによる生産管理システムのシュミレーションモデルの開発	
	ETC導入前の総合試験	
	OracleMaster: eラーニング	
	Si基板にイオン照射を行う効果	
	Windows2000によるLAN構築	
	パナソニック社内LANの見直し	
	ビデオインテックスの作成	
	ホームページの作成	
	メールングリストの構築	
	顧客管理	
	信号処理	
	通信工学関連分野の実験	
	半導体生産ラインにおける設備管理システムの調査、開発	
	放送通信融合サービス提供	

### 3-2-1-d-2. インターンシップへのご意見

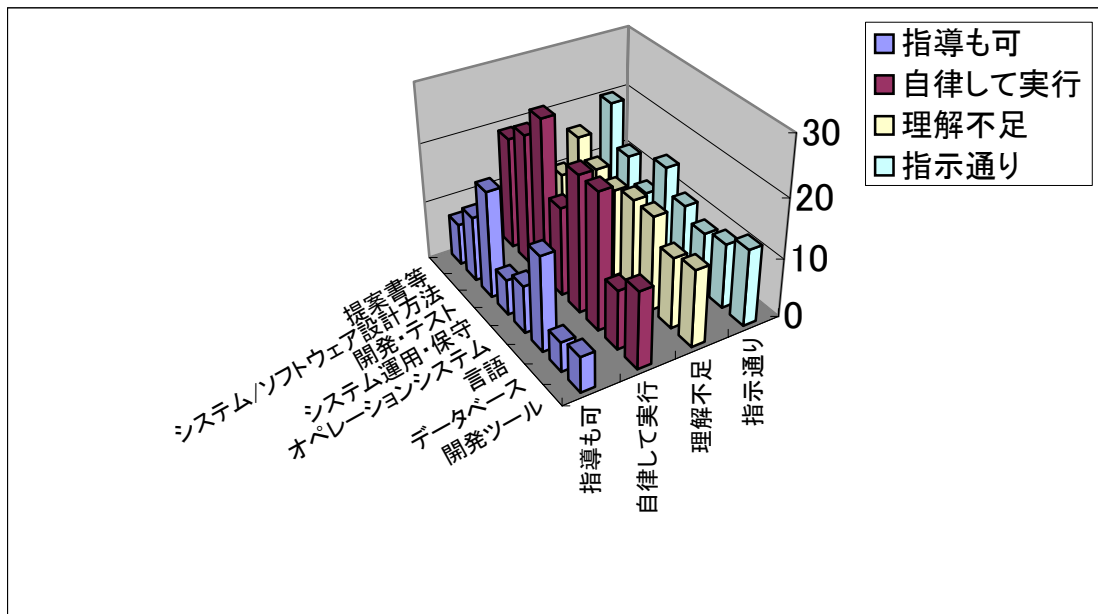
意見-分類	意見
現場の雰囲気を感じられる	<p>グループで1つの課題にとり組む市井は、社会人になった後も、有用であるので、より積極的に取り入れるべきだと思う。</p> <p>プロの開発者の中での実際の作業を経験することにより、何を学ぶべきなのかが少し分かる。</p> <p>学生のうちに企業での研究風景を見ておくのもとても重要だと思います。企業の雰囲気を学生で感じることができ、貴重な体験ができた。期限や品質に気をを使う必要があり、責任を持って仕事を行う事の難しさを知ることができる。</p> <p>工場実習として、計算機メーカーに3週間お世話になったが、現実の職場の雰囲気を感じることができ、非常に有意義でした。今後より盛んになっていただきたい教育手法と考えます。</p> <p>実際に会社に行くことで、就職する前に会社の雰囲気を味わうことができる。がプロのSE上から指導を受けることができるので、最新の技術にふれることができる。(大学で教わるものは少々古い気がする)</p> <p>実際に職場で働くことを経験できるので、“仕事”を知ることができるため、良い精度だと思う。</p> <p>実際の実務や社員の話しが聞けるチャンスなので、大変有効</p>
研究の刺激となった	<p>プログラミングについて新たに学べた点が多く、その後の研究にも役立った。</p> <p>実際の業務を体験できることで、勉強へのモチベーションがあがる。</p> <p>大学から離れたところで、またいつも会う人と違う人と会うことで、研究意欲が高まり、そこで働く人の話が聞けるので、素晴らしい効果があると思う。</p> <p>大学で学んだ知識に対して、実際の現場で応用していくためには、どのような観点での改善・工夫・進歩が必要なのかということ深く考えさせられ、その後の研究の刺激になった。</p> <p>短期間ではあるが、社会に出て時際に社会に「もの」を送り出している人々と接することは、研究室の中では感じることはできない大きな責任感を感じることができ、その後の研究して、ただ研究するだけで歯なく、社会に通じる研究という気持ちで取り組めた。</p>
大学教育の無意味さを感じられる	<p>大学の教育には効果なし。その職に就く場合は効果あり</p> <p>非常に重要。企業に出す人材を育成するために大学は機能していない(研究者のための教育となっている)</p>
やりたいことが明確になった	<p>実務につく前に、実際に企業で行われている作業を体験できるため、今後の自分の目標、やりたい仕事内容等を明確にする手助けになる。</p>
意味がない	<p>擬似的なインターンシップはあまり意味がないと思う。「社会人ごっこ」インターンにしても、特にカルチャーショックを受けることもなかった</p>
その他	<p>自主的に働く/積極的に技術を学びに行くという姿勢が重要。受身的義務的なインターンシップは会社に迷惑。・アウトプットを出し、そのアウトプットが社会で生きるという経験は貴重。それはエンジニアの自信となり、好奇心を育てる。</p> <p>HTMLの基本的なことは覚えられたと思う。しかし実務で学んだ方が身につくのは否めない。</p> <p>インターンシップの結果を発表する機会などがあれば良いかと思います。成果を基礎知識のない人へ判り易く説明するためには、どのようにすれば良いかを学ぶことができた。</p> <p>有意義で、もっと行うべき</p>

### 3-2-2. 習熟度について

- 3-2-2-a-1. 項目毎の習熟度
- 3-2-2-a-2. 開発プラットフォーム
- 3-2-2-a-3. 規模（人月）
- 3-2-2-a-4. 規模（人数、期間）

### 3-2-2-a-1. 項目毎の習熟度

	習熟度			
	指導も可	自律して実行	理解不足	指示通り
提案書等	7	19	10	20
システム/ソフトウェア設計方法	11	22	19	13
開発・テスト	18	27	16	9
システム運用・保守	6	15	15	16
オペレーションシステム	8	23	16	12
言語	16	23	16	10
データベース	5	10	12	11
開発ツール	6	13	13	13



### 3-2-2-a-2. 開発プラットフォーム

	1回目	2回目
パソコン	29	2
ワークステーション	32	3
メインフレーム	0	0
その他	0	0

3-2-2-a-3. 規模(人月)

人月	一回目	二回目
0	49	76
1	8	1
2	3	
3	5	
4	4	
5	1	
6	2	1
12	1	
15	1	
平均人月	3.6	3.5

3-2-2-a-4. 規模(人数、期間)

規模		一回目	二回目
人数(人)	期間(月)		
0	0	56	76
	3	1	
1	1	1	
	6		1
2	1	2	
	6	1	
3	0	1	
	1	2	
	2	1	
4	3	1	
	0	2	
	1	1	
5	3	1	
	0	1	
	1	1	
6	0	1	
8	6	1	
20	3	1	

規模	
一回目	4.1人 : 2.4ヶ月
二回目	2人 : 6ヶ月

※無記入(0)を除いて算出

### 3-3. 授業以外でのシステム開発経験について

- 3-3-1-a-1. 経験について
- 3-3-1-a-2. 「その他での開発経験」の記入
  
- 3-3-2-a-1. 習熟度について
- 3-3-2-a-2. 「開発プラットフォーム」の習熟度  
について
- 3-3-2-a-3. 開発の規模
- 3-3-2-a-4. 開発内容

3-3-1-a-1. 経験について

	クラブ活動	アルバイト	その他
クラブ活動	22	10	3
アルバイト	2	81	5
その他			33

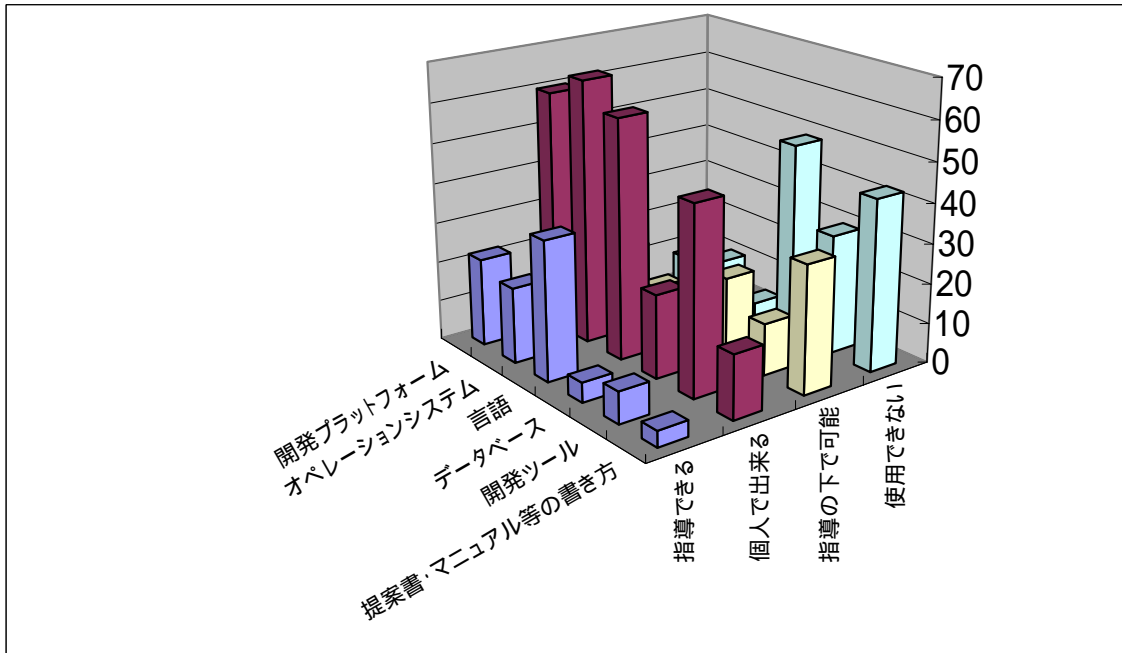
※システム開発の場合(クラブ活動、アルバイト、その他)についての経験数

3-3-1-a-2. 「その他での開発経験」の記入

その他での開発経験
卒業のための開発・自分の勉強、テスト用の開発
ゲームのプログラミング
ネット上でのフリーウェア開発
フリーウェア
フリーソフトウェアを配布
フリーソフトの作成・配布
プログラミング雑誌への投稿
モトローラMC6809用のクロスアセンブラの試作
ロボット制御でプログラム自作
起業経験あり
共同研究における開発
研究論文にしようとした数値計算のために、Fortran, C言語を独学
研究論文作成時
個人
個人的なプログラミング経験
個人的な開発
個人的に
個人的興味
自らの興味で
自作ソフト(ゲーム)の販売フリーソフトの開発
自宅での独学
趣味
趣味で
趣味での開発経験あり
趣味で個人的にプログラム作成
趣味にて、研究にて
卒業研究
大学・大学院での研究
独学
父のインターネット開設支援

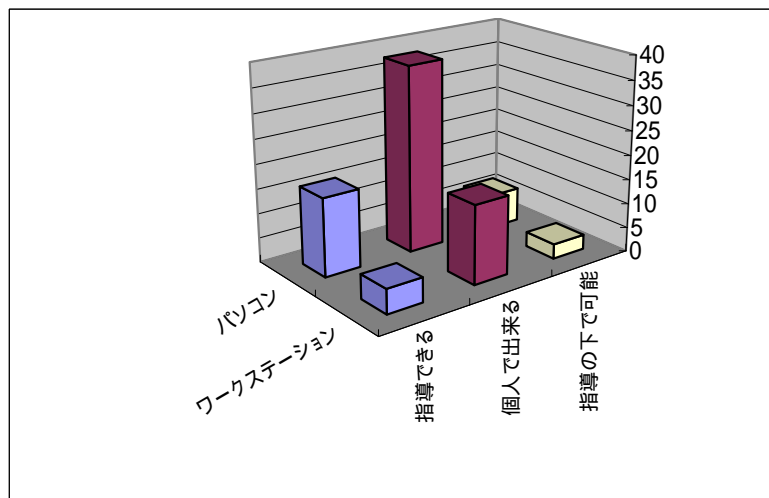
### 3 - 3 - 2 - a - 1 . 習熟度について

	指導できる	個人で出来る	指導の下で可能	使用できない
開発プラットフォーム	22	60	15	8
オペレーションシステム	19	66	9	11
言語	35	60	7	4
データベース	5	21	20	49
開発ツール	8	47	13	30
提案書・マニュアル等の書き方	4	16	32	43



### 3 - 3 - 2 - a - 2 . 「開発プラットフォーム」の習熟度について

開発プラットフォーム	指導できる	個人で出来る	指導の下で可能
パソコン	16	38	7
ワークステーション	5	16	3



### 3 - 3 - 2 - a - 3 . 開発の規模

平均規模(人月)	4.3
----------	-----

### 3 - 3 - 2 - a - 4 . 開発内容

分類	テーマ	数
専用アプリ	CMSの開発	26
	PC用ソフトウェアコピープロテクト作成	
	X68000での、アプリケーション開発	
	サーバーアプリケーションの開発	
	システム開発	
	タイピングソフト	
	データベース作成、支援ソフト作成	
	パッケージソフト開発	
	プラグイン開発	
	ヘルプディスクシステムの開発	
	ホームページ、改ざん検知システム	
	マルチメディアプレイヤー	
	画像データベースの管理	
	学習塾の成績処理シート作成	
	減色エンジン	
	出玉管理等	
	出席管理DB	
	新聞店広告折り込み管理システム	
	人材採用システムの開発	
	専用アプリ	
不動産、住宅ローン計算システム		
不動産管理		
複数端末間通信インターフェース		
文化施設検索システム		
麻雀プログラム		
力覚デバイスにおける人間知覚の研究		
Webアプリ	CDリスト管理CGI	11
	Javaを使用した、バス路線図提供システム	
	Web & 事業システム	
	Webアプリ	
	Webアプリケーションシステムの開発	
	WEBからメールを読むシステム	
	webの購買サイト	
	Webベースのメールリストマネージャ	
	Web上で作動するゲーム	
	検査用CGI作成	
専門学校生徒の授業管理		
ゲームソフト	ゲーム	9
	ゲームソフトの開発	
	ゲームソフトの開発	
	ゲームのプログラミング	
	ゲーム制作	
	パズルゲームの開発	
	宿祭バトル	
	将棋ソフトの開発	
心理ゲーム(占い)		
帳票印刷ソフト	帳票印刷	4
	帳票印刷モジュール開発	
	帳票印刷機能	
環境構築	伝票出力	3
	Linuxサーバの構築運用	
	Webサーバの構築	
	インターネット開設支援	

分類	テーマ	数
地図ソフト	数値地図のビューアー、PS形式での印刷ツール 地図データ作成システム	2
言語処理系ソフト	インタプリタ モトローラMC6809用のクロスアセンブラの試作、設計	2
携帯ソフト	ビルメンテナンスツール 開発(携帯機器)	2
教育用ソフト	M塾の通信教育ソフト インターネットを利用した学習システム	2
解析プログラム	医学向け解析システム 微分方程式の解析プログラム	2
画像処理・認識	画像処理 SVM(サポートベクタマシン)を用いた画像認識 画像動画処理	3
その他	論文用資料統計ツール サーバ管理 Webサーバ管理プログラム GPSソフト用GUI製作 移動ロボットの遠隔操作及び位置の画面表示 ホームページ作成 計測結果のLogging&Viewer 小規模DBとインターフェイス構築 ライン制御システム プログラム開発、教員 CPUコア設計・アセンブラ設計 システム間連携 アルゴリズム解析	15

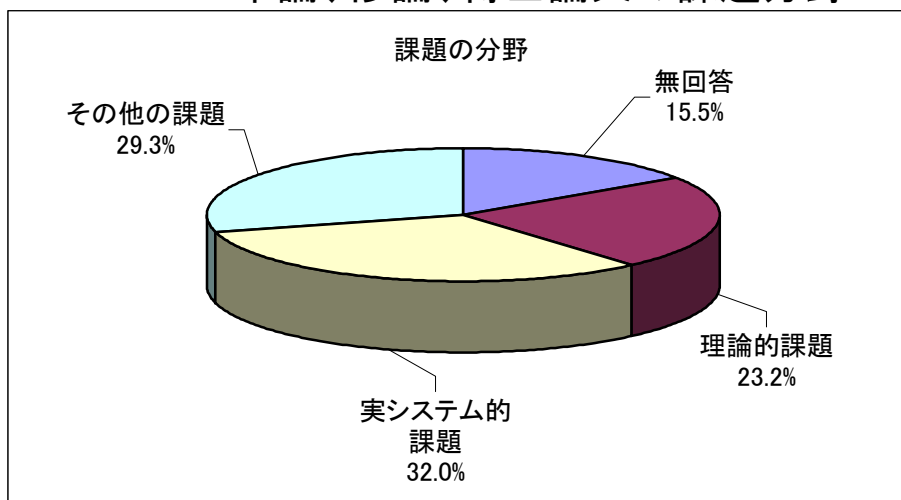
### 3-4. 卒論・修論・博士論文について

3-4-a-1. 卒論、修論、博士論文の課題分野

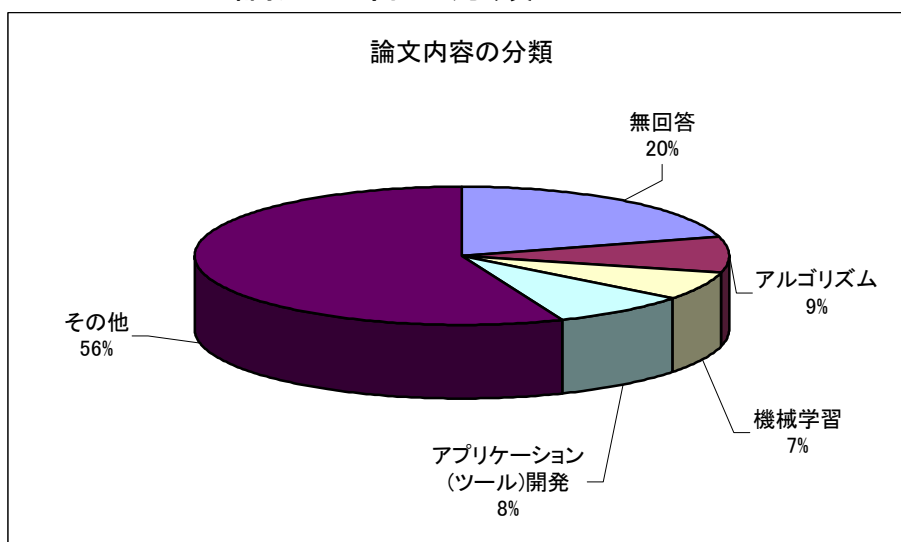
3-4-a-2. 論文内容の分類

3-4-a-3. 業務で役立っていること

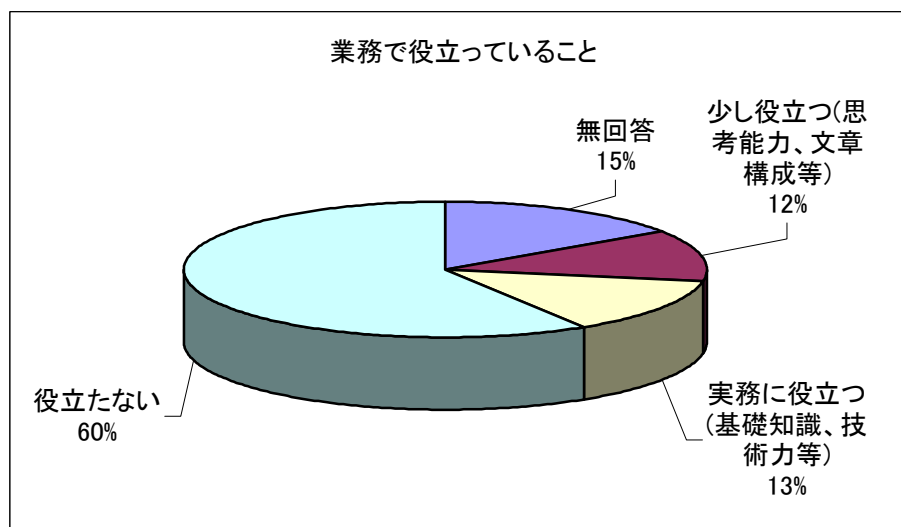
### 3-4-a-1. 卒論、修論、博士論文の課題分野



### 3-4-a-2. 論文内容の分類



### 3-4-a-3. 役立っていること



### 3-5. 学会、論文の経験と効果について

3-5-a-1. 学会の経験数

3-5-a-2. 教育効果

3-5-a-3. ご意見の分類

3-5-a-4. ご意見の詳細

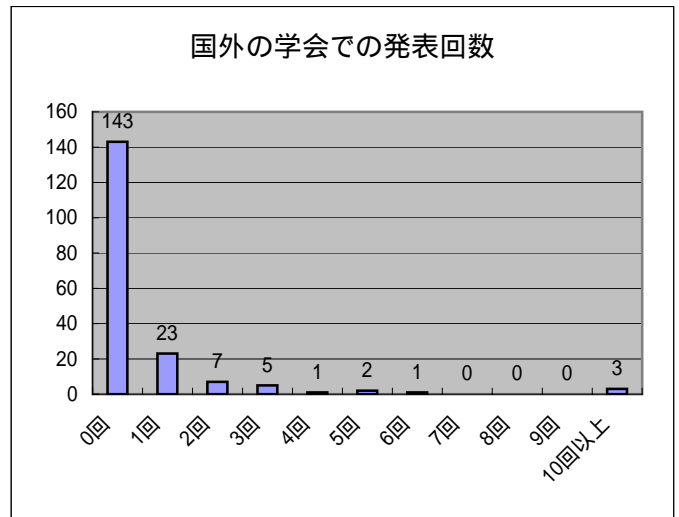
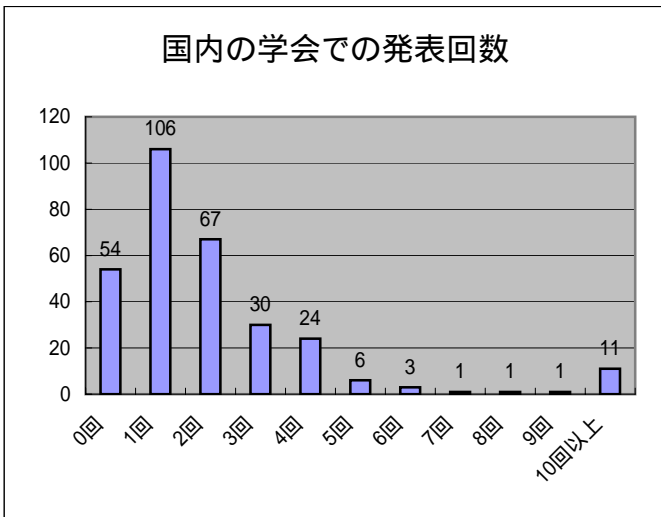
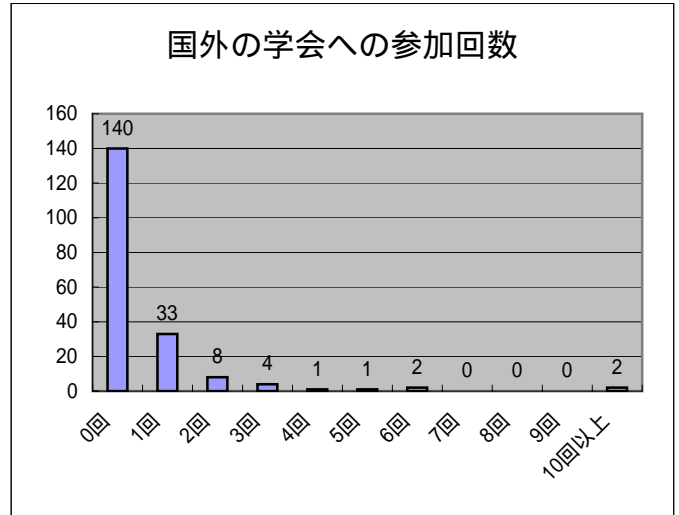
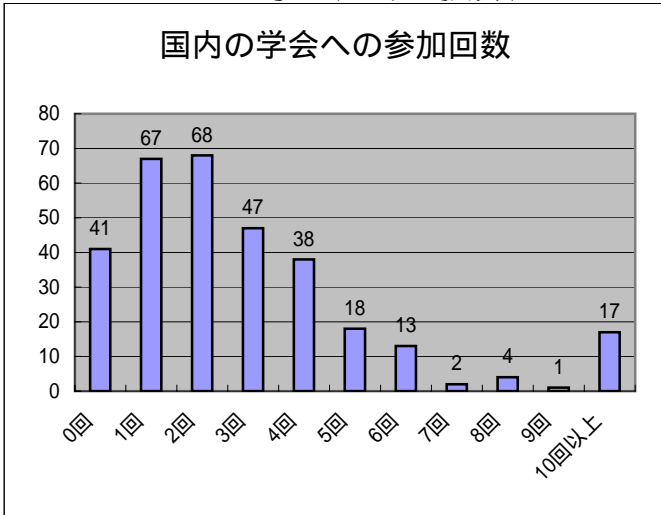
3-5-b-1. 論文の経験数

3-5-b-2. 教育効果

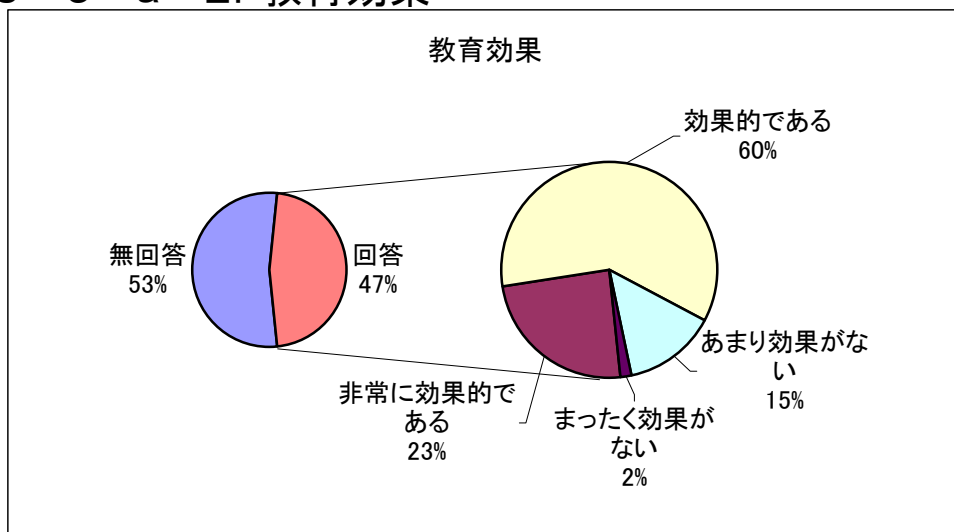
3-5-b-3. ご意見の分類

3-5-b-4. ご意見の詳細

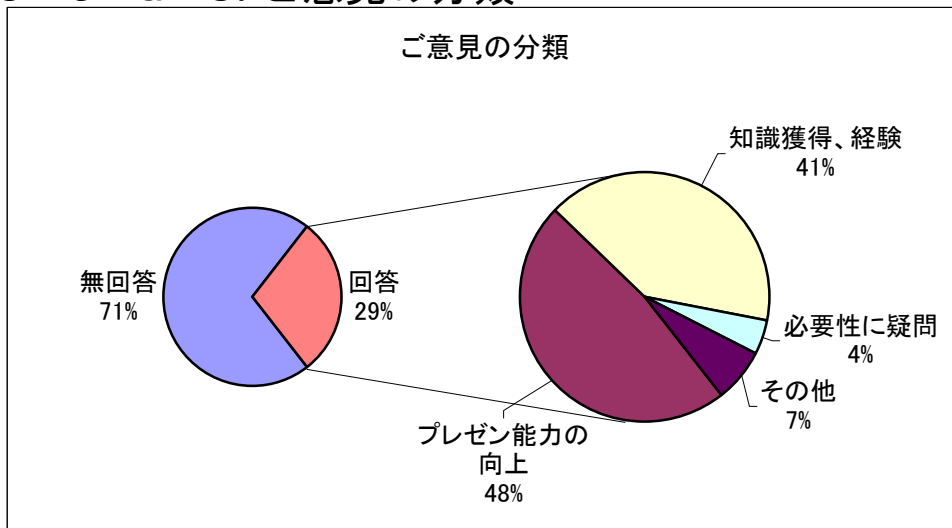
### 3 - 5 - a - 1. 学会の経験数



### 3-5-a-2. 教育効果



### 3-5-a-3. ご意見の分類



### 3 - 5 - a - 4 . ご意見の詳細

学会参加への具体的意見
・プレゼンテーション技能が向上すると思う。 ・他の研究からインスパイアされる
・主張することの難しさを学ぶ。一言で分かりやすく“従来との差”を説明するノウハウは一生涯必要。 ・国際学会を目標とすると、大きな研究をする意志が強くなる
・情報交換の場 ・プレゼン学習
・他分野の知識が身に付く ・最先端の技術(流行っている技術)についての知識が得られる。
・短い時間の中でないようを簡潔に分かり易く説明すること ・他者からの質問・意見に対する対応上記2点は、コミュニケーション・スキルを身につける効果がある。
・発表となれば、いい加減なものは作れない。 ・他の発表者から刺激を受ける。
q術文書を書く技術が身につく
その後研究をするにあたり、経験として役立つ
その分野に関する体系的な知識が身につくから。
なし
ノルマ的な参加には全く意味がない。ただ時間を無駄にしている。
プレゼン、ノウハウ、他者の研究内容については実はそれほどでも、という気がします
プレゼンスキルが向上する
プレゼンテーションスキルがつく
プレゼンテーションになれる。査読者の意見が聞ける。
プレゼンテーションの訓練として役に立つ。
プレゼンテーションの実践練習になる
プレゼンテーションの方法などは学べて良かった。
プレゼンテーションや他の研究者との質疑応答企業に入っても役立ちそうだが、適切な指導がないため、あまり効果がなかった
プレゼンテーション技術の向上、成果を期間内で上げる方法、新規の課題に対してのアプローチ方法などを身につけるために効果的。
プレゼンテーション技術を学ぶ良い機会
プレゼンテーション技法が格段にのびると思います。
プレゼンテーション能力が高まる。質疑対応ができる。
プレゼンテーション能力が身につく。自分の研究や考えていること的位置を知る場になる。
プレゼンテーション能力の向上
プレゼンの仕方
プレゼンの質が高められる最も重要な機会です。アメリカに負けているのはプレゼン能力であるといつてよいので
プレゼンの練習になる
プレゼンは現在でもよく行われている。プレゼン技法を学ぶ上でも有用である。
プレゼン能力がUPするから良いと思う
プレゼン能力の向上
プレゼン能力等
プレゼン力の強化
まず、その分野でどのような研究が行われているか知るために有効である。また、発表を行うことで、プレゼンテーション能力向上も見込むことが出来る。
もっと早い時期から卒論のような割と大きなプロジェクトをやっておくべきだったと思う。
よい経験となった。
何でも出せばいいというものでもないが、プレゼンの練習程度になる
会ったことのない人達と、同じテーマで議論できることは、有意義であると思う。
皆の前で、発表する練習になる。
外の人と触れ合うのは大事
外部からのコメント・批評を受けられること。
外部への発表を行うためプログラミングや実験結果を他人に理解できるように整理しなければならない。少人数で研究を行っているとしても他者とのコミュニケーション能力が低下するので定期的に発表は出来るだけ行った方が良い。
学会での発表準備等で、研究の進行が妨げられるが、今までの研究をまとめ、他の人に理解してもらう事は重要である。
学会という自校外の場で発表する事で、自分の研究を客観的に評価してもらう事ができる体験は、学生のうちに経験すると良いと思う。

学会参加への具体的意見

学会の参加は、国内外の先輩技術者・学者からの厳しい意見を得られるところでもあるし、広い視野で自分の研究を見つめ直すことが出来ると思う。特に社会人技術者との交流は、学生時代には通常得られない経験を得られると思う。

学会の参加は、多角的な議論の中で、知識を高めるのみならず、プレゼンテーション手法や研究意欲も向上させることが出来る点で効果的である。

学会の場での他の研究者の意見には大変助けられた。

学会の雰囲気「ゆるい」場合が多く、学会に参加する効果は感じません。

学会への参加に教育効果を求めているのは本来転倒、学会は学の探究のためにありそれ以上でも以下で学会をひとつの区切りとして、自分の研究をまとめていくことで、成果の確認や研究の進め方、進捗度を確認出来る。自分の専門のみならず、周辺の最新研究にふれることが出来る。自分の研究のヒントになるようなことに気付くことが出来る。(これは分野にとらわれず、いろいろな学会をみれるとよい。)研究会やワークグループの発表も同様のことが言える。発表することに関しては、プレゼン能力の訓練にもなる。

学会参加により、最先端技術が分かる。

学会発表することによるモチベーション向上

学会発表にて、学生を試すような質問が目立つ。アドバイスも同程度あるべきでは？

学会発表の準備過程で自ずと勉強する

学会発表もノルマ的なものがあり、発表のためにやっているという感じもあった。

学外の意見を確実に取り組める

学者を目指す場合は効果があると思うが、ITスキルを身につけるためにはほとんど関係がないように思われる。

学生時代に数少ないプレゼンテーション能力を養う場であると思う。

学生生活において、発表の機会を得ることで、プレゼンテーションの難しさを実感できた。

関連分野の動向を理解する機会となる

企業との交流、研究水準の認知

期日までに具体的な成果を修める必要があるため、それに向けた努力ができること。

技術動向調査や、他分野との交流が特に有効的

客観的な視点から見た意見がもらえること

客観的評価(自己の研究について)、情報収集、社交性の増加、プレゼン能力の向上、文章作成、文章力の向上、異文化を知る

教官が学生任せにしてあまり指導しないのは良くないです。授業料に見合った指導をすべきと考えます。

経験になる

経験はありませんがプレゼンテーション能力が身につくことが期待できます

研究していることに対して意義と責任をしっかりとつことが出来る。

研究について、広い視野が身に付く

研究に身が入る。

研究の目的や必要性について再度見直すべきでは(発表の必要性も)。学会は学生の発表練習の場ではないと思う

研究時には専門的な視点で深く掘り下げた見方をしてばかりいることが多いのですが発表の場では分野は同じだとしても専門の違う方に理解させる必要があります。特に、外部の方の視点は現場で最も気を付けている部分とは別の点(特に研究に至った経緯)に注がれますので自身の勉強の整理や背景の記憶を行うよい機会となると考えています。

研究自体は教育的な効果は大きいと思う。ただし直接的に業務に役立てる職場が少なく、需要もない。研究職ではなくても、技術開発職という専門職があり、需要が高くなれば研究は重要なものとなると思う。産と学のギャップが大きい。

研究室内や学内だけでなく、自分の研究が社会的にどのような評価を受けるかが確認することができ、より良い成果を出すことができる。

研究者と呼ばれる方や、他大学の学生・教授との接点となる。外側で起こっている出来事に感心を持つきっかけとなる

研究発表を設けることで、目的・目標を明確に出来る。発表の場で、色々な意見をきくことで、今後の研究の参考になる。

現状では修了条件という意味合いが強い

限られた時間のなかで自分の研究成果を発表しなければいけないため論点をまとめる力や、またそれを人に対して話すという力が付き実社会における交渉やプレゼンテーション等に役立つと思います。

公の場で、全く知らない人達への発表なので、プレゼン能力の向上があると思う。

公の場で発表するという事はいい経験になる

学会参加への具体的意見

考えや知識をまとめ、発表することで、現状の理解をさらに深めることになる。  
国内の学会は発表のレベルが低く、国際学会は英語が聞き取れなくて効果が低かった。  
国内の大学が企業が現在どのような事を研究しているかを見るよい機会である。  
今までの成果を分かり易くまとめる技術。プレゼンテーションの実戦への場。  
最初が肝心、テーマの見つけ方から研究内容、発表までしっかり指導する必要あり  
最新の情報系に関する知識を身に付けられる  
仕事の成果をまとめて、文章を作成し、発表を行うことは全ての基本  
私は行ったことがないが、非常に良い経験になると思う。  
視野を広げ、考え方を進歩させた  
資料のまとめかた、発表のしかた  
資料をまとめたり、短い時間でプレゼンを行う能力が鍛えられる  
自ら研究スケジュールを立て、そして実行していくこと。つまり、自主性を養うことができる。  
自分が証明しようとしている内容を、限られたスペース・時間でまとめあげる能力を養える。  
自分で作成したシステムに対して、他人からの意見を聞くことができる。また、同様のシステムの実現方法の  
違いなどを知ることができる。  
自分とは異なった視点での話を聞くことができる。視野が広がる。  
自分のやっていることを他人に理解してもらうための工夫。他人のやっていることを理解しようとする努力  
自分の研究(知的資産)をまとめて、発表する能力が培われると考えます。自分自身はやったことがありませんが、やっておけば良かったと考えています。  
自分の研究が行き詰まった時に、他の研究者の発表を聞く事は、何かのヒントになる事があると思います。  
自分の研究をまとめる・客観的に見れるよい機会  
自分の研究を人に伝えるプレゼンというものの体験が社会に出ても非常に役立つため  
自分の研究を他人にわかりやすく説明することで、自分の中で何をりかいしていないかがわかる  
自分の研究を他人に説明する事を通して、自分の研究の位置付けと方向を再確認できることが有効  
自分の研究成果を他の人にわかりやすく伝える良い訓練になった。また、他の研究機関や企業の方々の発表を聞くことでよい刺激を受けた。  
自分の研究内容について多くの意見を聞くことができ、意識の向上や自信につながる。また他の人の発表を聞くことにより、新しい発見があったり、技術的の動向を知ることができる。  
自分の研究内容や研究テーマについての知識や理解が深まる。  
自分の研究内容を多数の特別に専門でない人に対しても説明することで、プレゼン能力が向上、また整理に役に立つ。  
自分の言いたいことをうまく相手に伝えて説得する能力は、技術者にとって重要、かつ、不足している能力と  
思っていますが、その能力の向上にとって、学会での発表はよい機会だと思えます。  
自分の考え(技術)を、初対面の人に伝える難しさを実感できる。また、言葉が違って、考えのポイントは同じであることを実感できる。(国外)  
自分の実験や、研究によって得た成果を効果的に人に伝えるための訓練として、非常に有効であると思えます。  
質問への受け答えや、他の研究者らの発表に対しポイントを絞って聴く力を養うのには絶好の場である。  
少しでも発表の仕方やプレゼンの作り方、文章の書き方を学んでおかないと、社会に出てからどうしようもない下手な発表や分かりにくいプレゼン資料、分からない文章を書いてしまうので、そういった訓練のために必要と思われる。  
情報系とは関係ありませんでした。  
情報収集や人脈づくりでとてもよいと思う  
新たな研究テーマの発掘  
人に分かり易く話すのは簡単なようで非常に難しいため、その良い練習になると思います。  
人の研究について学び、質問、疑問をもつことは勉強になり、また違った考え方を学べる。、また、自分の発表をすることにより確実な理解を得ることができる。  
人前で発表することで、自分の考えを理解させるにはどうしたらよいかを学ぶことができる。  
人前でのアピールの方法、他人への自分の考えの伝達方法。  
人前でのプレゼンテーション能力が試されるため、発表の流れや説明方法について学ぶことが出来る。  
人前で自分の考えを説明する能力が身につく。人に見せる資料を作ることができる。  
人前で発表することへの慣れ。  
人前で発表すること自体、自分の理解度も客観的に分かるし、(質疑に対応しているとよく分かる)自分の考えで人に伝える、プレゼンもよい練習になる。  
人前で発表する練習になる  
人前で発表を行うということが重要な経験

学会参加への具体的意見

人前で発表を行うということで、非常に緊張感を持って研究にはげむことができる。  
 人前で論ずる技術。質問に対して的確な回答を行う技術  
 人前で話す経験とその分野でのスペシャリストの話が聞ける  
 成果物の見せ方、出し方などを学べる。  
 生の経験を聞くことによって共感が得られ、興味範囲が広がる。  
 専門的な人が集まる場での発表には経験的なこともふまえて効果があると思う。  
 全国大会などで質の低下が自立つ  
 相手に自分の考えを伝える方法を学ぶことが出来る。  
 他の研究にふれる事で自分の視野が広がる。  
 他の研究より刺激や知識が得られる  
 他の研究者の研究に対する視点を知ることができる。  
 他の研究者の研究内容を知ることが出来る(ITに限らないが)プレゼンテーションの仕方等を学んだ。  
 他の研究成果を見ることは非常に貴重。研究室に入ってからでなく、一回生の時から頻りに参加すべきと考  
 える。  
 他の方がどのような分野に興味があり、着目しているのかがわかり、自分の視野が広がる。  
 他の方に、自分の考えを伝える能力の向上  
 他の方の発表及び自分の発表を通じて、プレゼンテーション能力がつく。他の方の発表を見ることで、外部  
 からの情報を取り込んだり、人脈をつくる事が出来る。プレゼンテーションは学会によく出ている人の方  
 が、一般企業の社員より上手であると思います。  
 他人が同分野で、どのような研究をしているのか、また他分野での考え方が参考になる。  
 他人が理解しやすい表現への配慮、秩序ある組み立てへの意識はもちろん、ふだんとは違った視点を持つ  
 人々との議論もあり、得がたい経験であったと思う。  
 他人に自分の意見を説明する技術が身につく  
 他人の研究発表を聴講することは、単に自分の研究を見直すきっかけになるだけでなく、プレゼンテーション  
 能力やコミュニケーション能力を身につける上で重要な機会であると思う。  
 他人の発表を聞いて、研究の動向をつかむことが出来る。  
 他組織での開発状況を理解でき、そこから自分の設計へのヒントを得られる  
 他大学・企業の発表が非常に刺激になる  
 他大学や企業の、同じような研究(アプローチや結果)を知ることが、自身の研究の今後に役立つ  
 多くの人に、何かを分かり易く伝えるには、どうしたら良いかを学ぶことが出来る  
 大きな刺激を受ける。発表に慣れる。人と知り合う。  
 大学院に入るまで、人前で発表する機会がなかったため、発生する事は、非常に効果的だったと思います。  
 大勢の前で発表することによる度胸が養われる。見知らぬ人と交わされる問答を通して、異なる視点から研  
 究内容を振り返ることができる。  
 大勢の前で発表することは、よい経験になったと思います(資料作成等含め)  
 単純に卒業のための発表というより、行って発表することが刺激になるという意味で効果的。なので、研究の  
 モチベーションが低い人へそれが義務化してしまい、あまり効果はないと思います。  
 担当教官との問題もあって発表の機会に恵まれなかった。(教官が途中で退職された)  
 同分野の研究者とのコミュニケーション・親交  
 同分野の他の研究を聞くことにより、知識を得ることができ、自分自身の研究の幅も広がるため。  
 特に企業の研究者による発表は有益であり、貴重な経験となった  
 発表、質疑応答の経験で、プレゼンテーション・コミュニケーション能力が向上する。  
 発表すること自体が経験になる。他大学の人の意見を聞ける。  
 発表する事により、内容をまとめる技術、人に伝える技術を習得できる。但し、技術作文を指導する教官は、  
 しっかりつべきだと思う。また、大会等に参加することで、さまざまな意見・考え・技術に触れる機会が多くな  
 るであろう。  
 発表の前にするプレゼン練習、質疑応答練習は、深い理解に繋がるだけでなく、人に理解してもらった技術が  
 身につく場であるので積極的に利用すべきである。また、学会の梗概集に連絡先を載せるだけでは生まれ  
 ない人と人の繋がりができるので、特に専門分野ではこのような場へ行くべきである。  
 発表の練習  
 発表は貴重な経験にはなると思うが、実務に関係無いことであれば社会では役に立たない。  
 発表をしなくても他の人と自分の意見の相異を気づくことや意見を交えることによって新しい発見、提案が生  
 まれる。  
 発表をする力、人に知ってもらった表現力がつく  
 発表経験。研究室以外の人との意見交換ができる。

学会参加への具体的意見

発表内容そのものだけでなく、当該分野での位置づけや意義といったことを理解することの重要が認識できた。論理的に説明する能力を身につけることができる。

文章をまとめる技術、プレゼンテーション技術に関してプラスになると考えています。

文章を書くことや人前で話すことは、とても勉強になった、と思っています。また、自分のやったことをまとめる練習にもなり、教育効果は高かったように思います。

文章理解能力の向上。標準的なコミュニケーション能力の獲得

文章力、表現力を身に付けることができる

様々な、分野の発表を聞くことによる、視野の拡大。

様々な意見・批判を聞け、また新しいアイデアに触れる機会がある。

様々な人が自分の問題点を指摘してくれるので学内で勉強するよりも少なくとも十倍は効果があります。

理学的な「本質」に対するアプローチが極めて薄く、民間の開発との差異が明確に認められない研究が多数ある。領域区分の明確化が必要ではないか。

理論武装のため、自分の中で疑問を組み立てたりすることが研究に反映されるのがよい。ふだん接していない協会の人などと8それがとんちんかんな質問であっても)会話することで新たに発見できることもあるかもしれない。

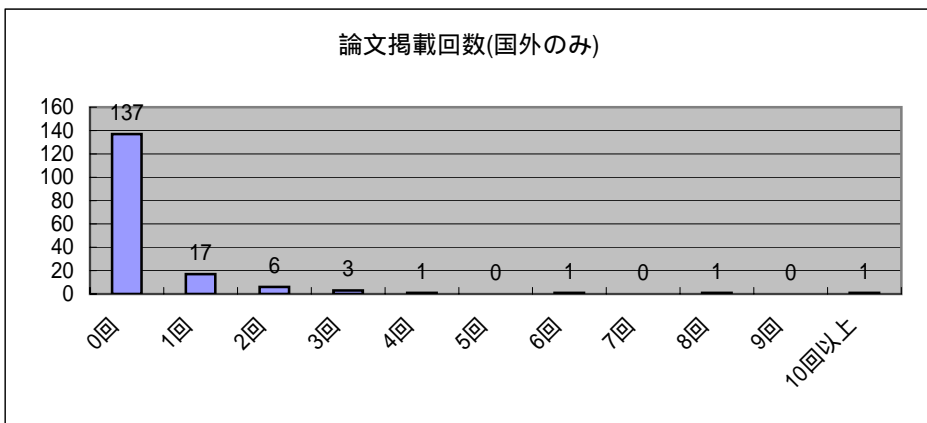
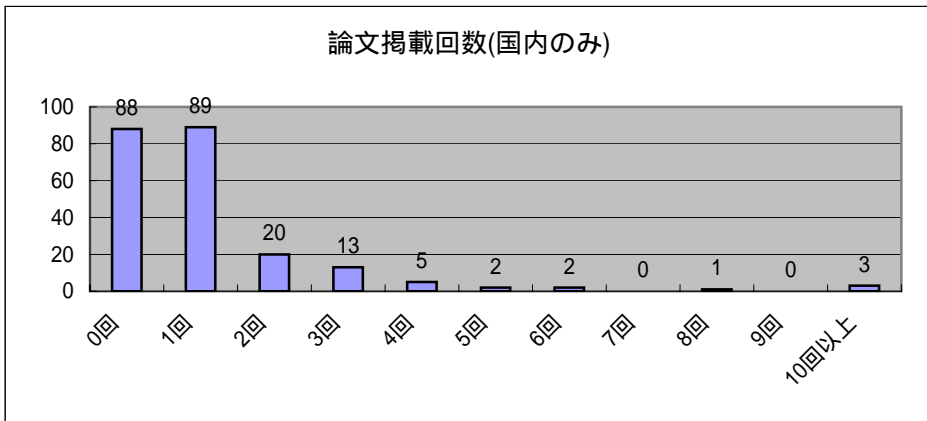
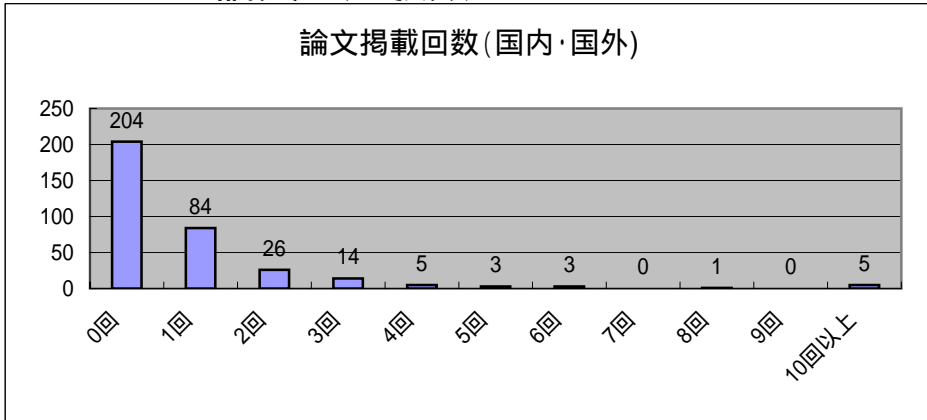
理論立てて発表する力が育てられる

論文という形にまとめる技術

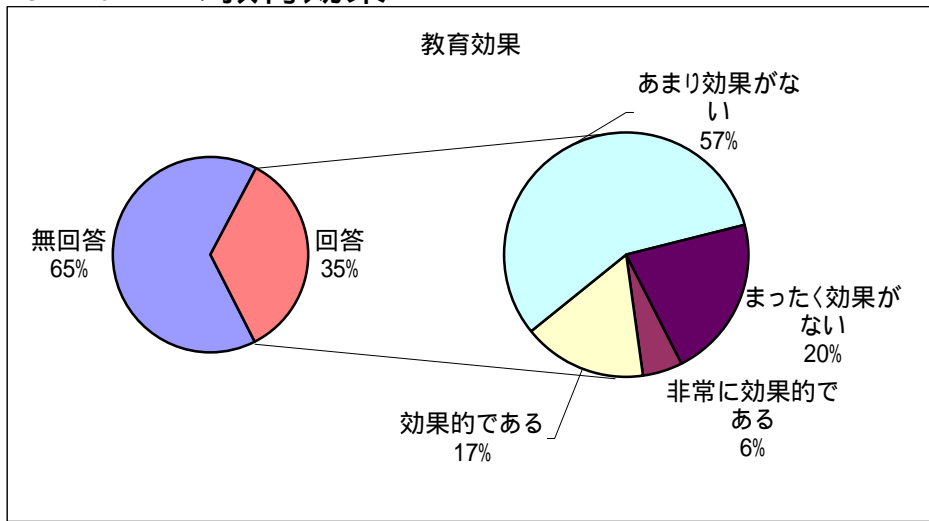
論文を投稿しても学会に行かれるのは、連名の教授が代表で行かれていたため、学会での発表経験はありません。

論理的な考え方や伝え方を学ぶことができた。

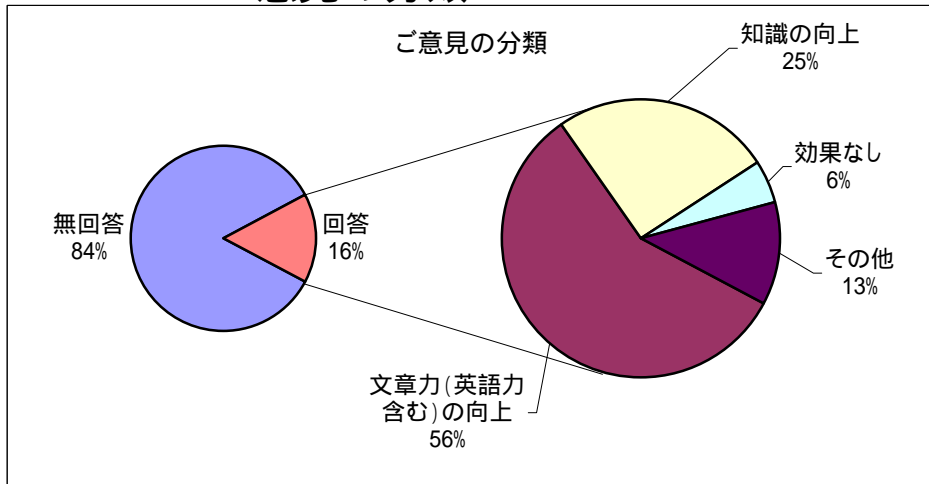
### 3 - 5 - b - 1 . 論文の経験数



### 3 - 5 - b - 2 . 教育効果



### 3 - 5 - b - 3 . ご意見の分類



### 3 - 5 - b - 4 . ご意見の詳細

論文掲載への具体的意見
文章作成能力が向上すると思う。
IT業界においても「設計書を書く」「マニュアルを書く」等、「文章を書く」という作業が重要となります。また、論文を書く事で論理的に考え、まとめるという力が付き実社会においての仕事に対する考え方に役立つと思います。
ただ出すだけの論文にならなければ効果あり
ドキュメント作成能力が向上します。
みずからの研究成果を広く公開し、意見をもらえる。
モチベーション向上
ものの書き方を学んだり、物事を完成させるためのプロセスを学ぶのに有益
よい経験となった。
英語での表現方法、明確な議論の方法を身につけるために役立った。
何かしらの意義をもたせる
会社でも論文を書く機会があるので、文章のまとめ方や課題の見つけ方等に役立っていると思います。
学会で発表するときも論文は書くので、特に論文誌向け書くことは重要ではないと思う。目標になるかもしれない
学会と同じ
学会への参加と同様論文をまとめることに、関しては技術文書の作成(作文)能力の訓練にもなる。
学会への発表と同じ理由です。
学生が主体で執筆する場合はとても効果があると思いますが、綿塩場合はそうでもないことが多かったです。今
から思えばとても残念
学内の発表とはまた違った視点で論文を作成する必要があるためよい経験であると思う。
企業で同様の機会があれば経験が役に立つと思います。
技術的な考察を深め、論理的な文章を作成する訓練になる。
技術文書の書き方の習得
技術用語の知識
客観的・論理的な文書を書く、プレゼンをする能力が身につく
客観的評価、論理的な文章作成、達成感
業務でも文書を書く機会がある為、論文の書き方を事前に学ぶことができることは有意義である。
掲載されるとやる気がてる
研究が理論的なものから、現実的なものに近づく。
研究のまとめ
研究者としての教育効果はあると思いますが、IT技術者としては学会への論文掲載に取り組むことが必ずしも効果
的とはいえないと思います。
効果的であると思いますが、経験が無いのでよき分かりません。
行っていないのでわかりません
査読に通るための論文を書くことは、説得において必要なデータを集める訓練になると思います。
査読のコメント対応のプロセスは、社会に出てからの説明、説得につながる
査読は、専門家が何を重視し何に価値を置くのか、肌で感じるができる重要な経験。
査読者の意見が聞ける。
指定された文字数で考えを効果的に伝える練習になりました。
視野を広げ、考え方を進歩させた
資料のまとめかた
自ら研究を文書化し、万人が納得できる結果を示すことができる。
自分のやったことが認められることなので、自分のやっていることに自身が持てると思います。
自分のやっていることを他人に理解してもらうための工夫。他人のやっていることを理解しようとする努力
自分の研究(知的資産)をまとめて、発表する能力が培われると考えます。自分自身はやったことがありません
が、やっておけば良かったと考えています。
自分の研究の振り返りと総括ができる
自分の研究を他人に説明する事を通して、自分の研究の位置付けと方向を再確認できることが有効
自分の行った研究内容をまとめるという作業が、自分の頭の仲を整理するという点でも有効であるし、文章をまと
めるという力の向上にもなる。
自分の頭の中が、一度整理されるので良い
自分は投稿したことがないが、技術文書の書き方が身について大変良いと思う。
自分達の行ってきた内容を、全国の学術研究者の方へ発表するための文章の書き方を学んだ。
順序立てて論ずる能力はベースとしてとても大切です。
情報が欲しいときはWebで検索する。検索した結果、論文に辿り着くことはあるが、明示的に論文を探すことはな
い。全文章をWebに掲載して欲しい。
情報系の学会は1つも有りませんでした。
人に理解するための表現力がつく
生果をまとめる能力、作図、作文、能力を高めることに役立った。
説明能力や論理能力の向上

論文掲載への具体的意見

先端技術のアピール及び、研究者としての地位の確立に影響大。幅広い意見を得ることが可能。  
他人が同分野で、どのような研究をしているのか、また他分野での考え方が参考になる。  
他人が読んで理解出来るように気をつけて、自分の中で整理する事が必要となり、それまでには、無かったので、非常に効果的だったと思います。  
大学で研究を続けるには論文を書く意味はあると思うが企業に務めるには必要ないものであり、いまに役に立っているとは思えない。  
大学の教授以外からの意見を受けることができたので、大変ためになった  
単純に卒業のための発表というより、行って発表することが刺激になるという意味で効果的。なので、研究のモチベーションが低い人hそれが義務化してしまい、あまり効果はないと思います。  
知識向上はもちろん、国外の論文では英語の学習にもなる。  
目的をとり違えたむやみな参加は学会、論文の質の低下をまねくだけ、日本語の勉強と研究についての位置付けを見極めることが可能となる。  
発表/参加の効果と同様  
発表したことがないので何もいえません  
発表内容そのものだけでなく、当該分野での位置づけや意義といったことを理解することの重要が認識できた。  
論理的に説明する能力を身につけることができる。  
文で問題を提起し、それを解決するにはどうすればいいか？考える力を養うことができる。  
文書を書くことも多いため、論文は有用であると思う。  
文書を書く練習になる  
文書作成能力の向上に役立ちそうだが、適切な指導がないため、あまり効果がなかった  
文書力の向上  
文章のまとめかた、という点での効果があると思いますが  
文章の作成は全ての基本  
文章の書き方、論理的な考え方の鍛錬。  
文章をまとめる技術、プレゼンテーション技術、人的ネットワーク構築に関してプラスになると考えています。  
文章構成力を磨くこと、論理的にストーリーを構築する力をつけることその他、査読者とのやり取りを通じた研究者としてのある種の社交性についての経験  
文章作成に非常によい機会である。英文で書くことでよりよい経験になる。また、自分に対する学術的な自身にも繋がる。  
文章作成能力が養える  
文章能力の向上  
文章表現能力が身につく。深く考える習慣が出来る。ただし、研究のための研究に陥る危険性があり、市場動向や社会的豊かさから離れた方向に向かってしまうものも敬具されるように感じる。  
文章力、表現力を身に付けることができる  
理論をまとめることで実用性へ結びつける  
理論を理解し、実験からのデータをもとに結果を導き出すことにより、理解が深まるため。  
論理的に説を組み立てて文章化する能力が身につく  
論文として成果を一つの書類にまとめることは、一生ついてまわるものである。例えその研究を社会に出たから行わないとしても、論文をまとめたことがある人と、そうでない人は社会人になってから明らかに差がつくことが多いと感じる。(会社で新人を見ていると感じる)  
論文の技術が得られる  
論文の作成方法自体の勉強になる  
論文の書き方が身につく  
論文を掲載することに重点を置き、内容が乏しいものが出てきて、本末転倒のようになりそう。  
論文を書く作業が論理的思考をよくする。  
論文作成はしていません。  
論文作成力及び英語力(重要!!)の意味より海外向きの論文作成はとても重要と考える。  
論文誌への掲載は、実務上あまり教育効果は無いと思われる。  
論理的な作文能力の向上  
論理的な思考・記述など  
論理的な表現力  
論理的な文章を書く練習になる。  
論理的に自分の研究を説明を行う必要があるため、自然に論理的な思考が身に付く。  
論理的に物事を記述する訓練になる。  
論理的文章作成能力の向上

#### 4. 企業内教育について

4-1-a-1. 期間について

4-2-a-1. 教育内容と習熟度について

4-2-b-1. 習熟したオペレーションシステム

4-2-b-2. 習熟した言語

4-2-b-3. 習熟したデータベース

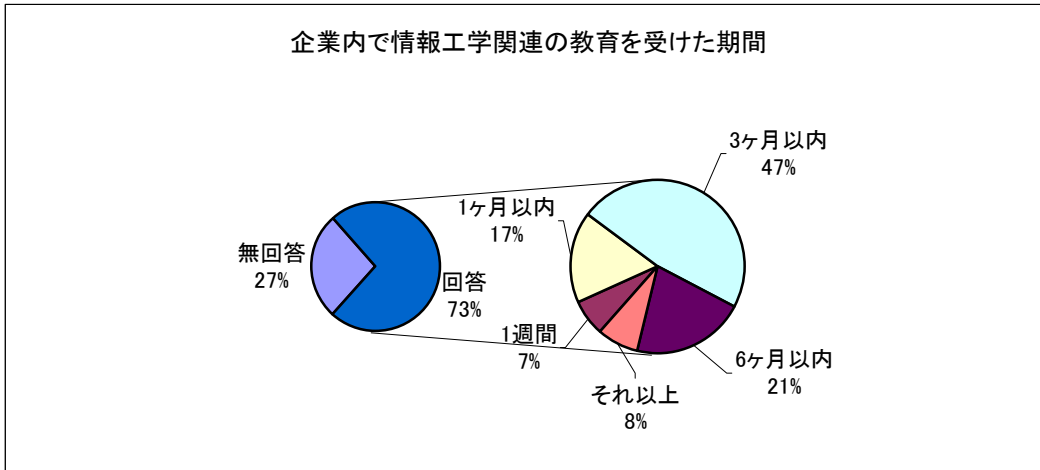
4-2-b-4. 習熟したアプリケーションサーバ

4-2-b-5. 習熟した統合開発環境

4-2-c-1. その他の学習項目

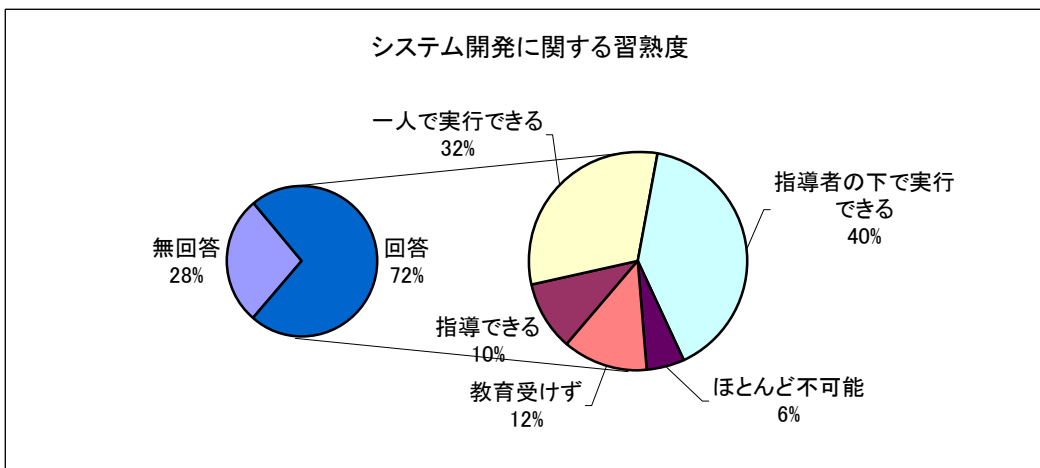
#### 4-1-a-1. 期間について

企業内で情報工学関連の教育を受けた期間	人数
無回答	176
1週間	31
1ヶ月以内	81
3ヶ月以内	224
6ヶ月以内	98
それ以上	36
合計	646



#### 4-2-a-1. 教育内容と習熟度について

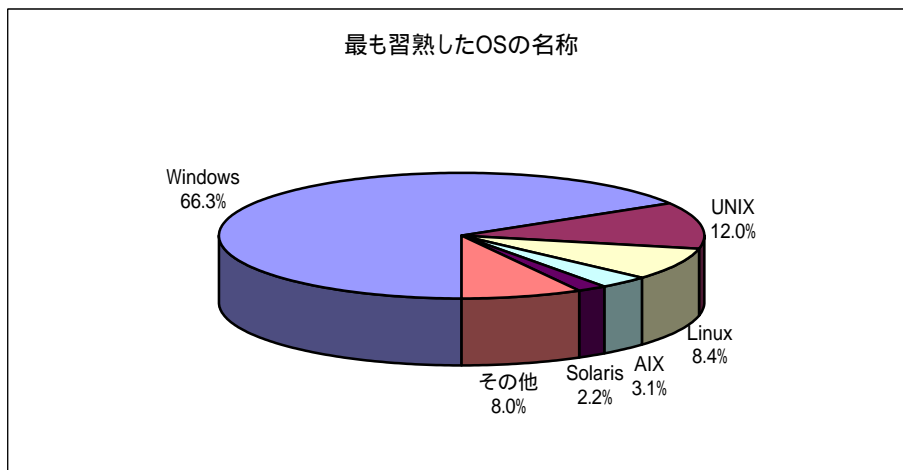
習熟度	人数
無回答	180
指導できる	48
一人で実行できる	148
指導者の下で実行できる	186
ほとんど不可能	26
教育受けず	58
合計	646



#### 4 - 2 - b - 1 . 習熟したオペレーションシステム

オペレーションシステムの名称(複数回答)	件数
Windows	275
UNIX	50
Linux	35
AIX	13
Solaris	9
その他	33

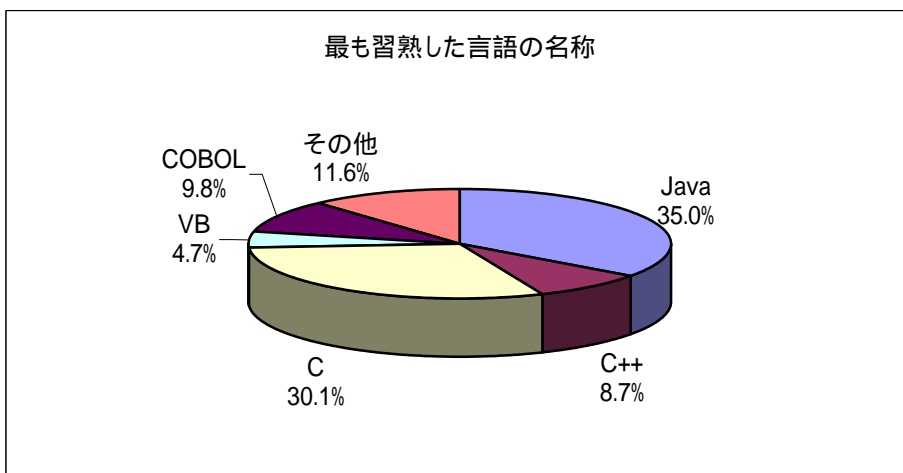
回答者
386



#### 4 - 2 - b - 2 . 習熟した言語

言語の名称(複数回答)	件数
Java	178
C++	44
C	153
VB	24
COBOL	50
その他	59

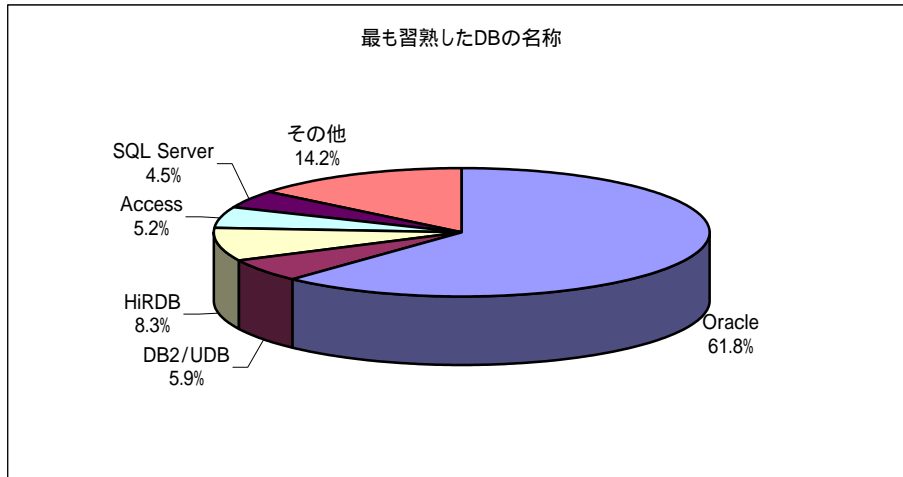
回答者
418



#### 4 - 2 - b - 3 . 習熟したデータベース

データベースの名称(複数回答)	件数
Oracle	178
DB2/UDB	17
HiRDB	24
Access	15
SQL Server	13
その他	41

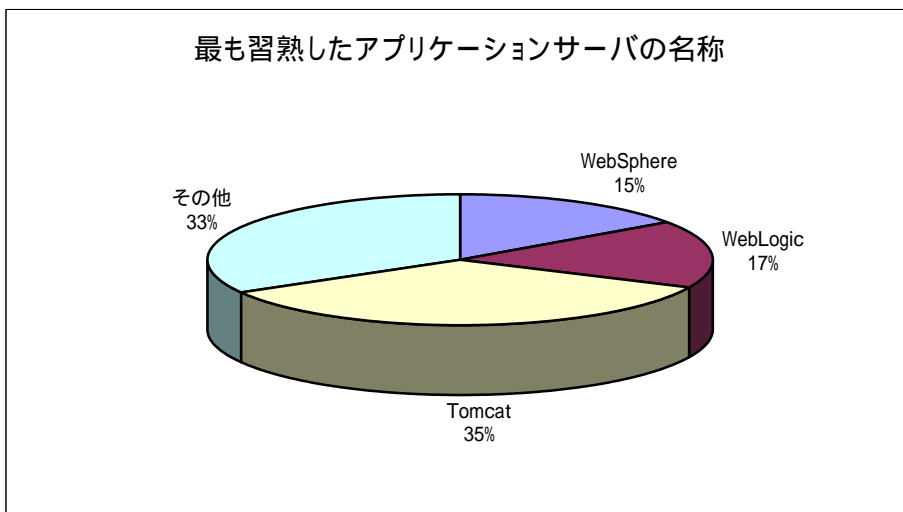
回答者
279



#### 4 - 2 - b - 4 . 習熟したアプリケーションサーバ

アプリケーションサーバの名称(複数回答)	件数
WebSphere	12
WebLogic	13
Tomcat	27
その他	26

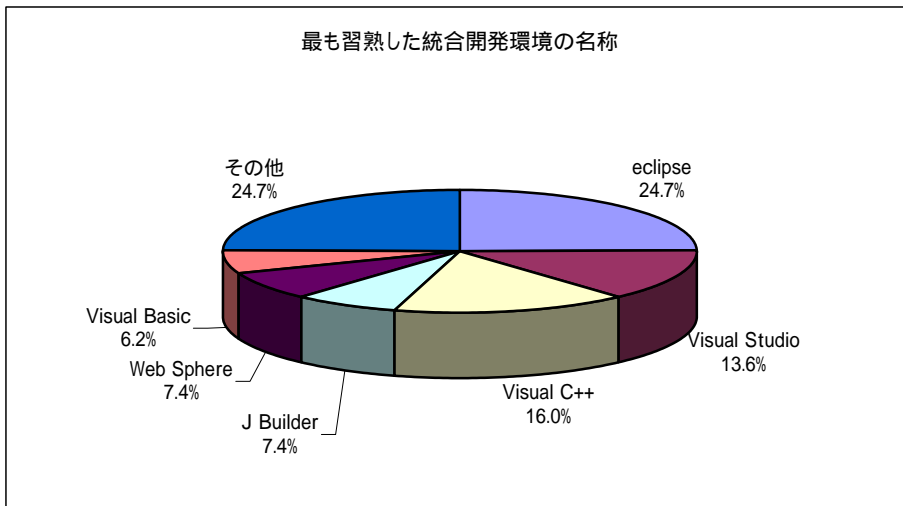
回答者
75



#### 4 - 2 - b - 5 . 習熟した統合開発環境

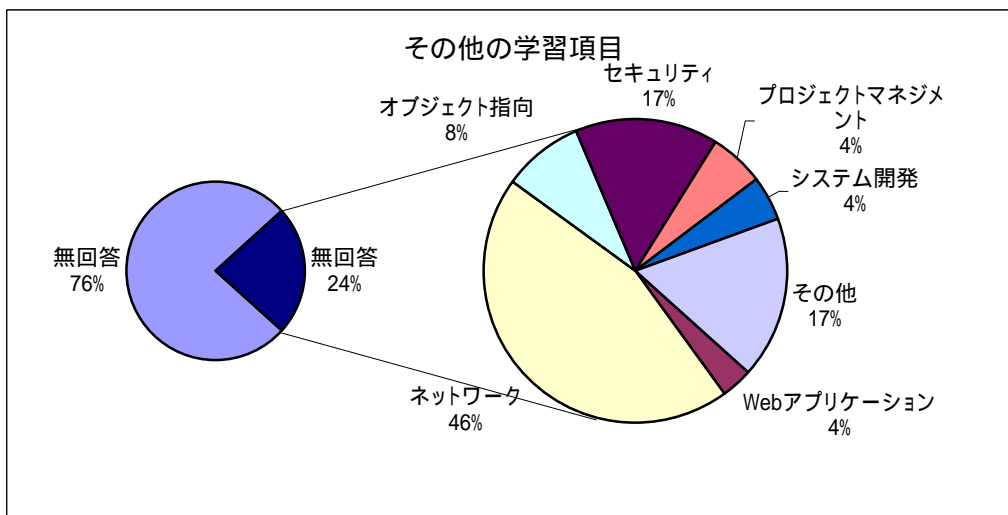
統合開発環境の名称(複数回答)	件数
eclipse	20
Visual Studio	11
Visual C++	13
J Builder	6
Web Sphere	6
Visual Basic	5
その他	20

回答者  
77



#### 4 - 2 - c - 1 . その他の学習項目

その他学習項目	件数
無回答	495
Webアプリケーション	5
ネットワーク	68
オブジェクト指向	13
セキュリティ	23
プロジェクトマネジメント	9
システム開発	7
その他	26
合計	646



## 5. 業務としてのシステム／ソフトウェア開発を通しての習熟度

5-1-a-1. 実際の案件での開発経験

5-2-a-1. 開発プラットフォーム

5-2-b-1. 習熟度：OS

5-2-b-2. 習熟度：プログラミング言語

5-2-b-3. 習熟度：開発技術

5-2-b-4. 習熟度：プロジェクトマネジメント

5-2-c-1. 開発時の特記事項の分類

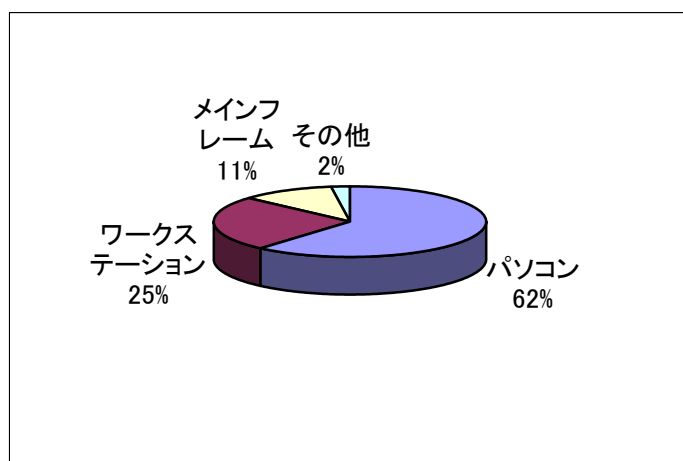
5-2-c-2. 開発時の特記事項の詳細

### 5-1-a-1. 実際の案件での開発経験

開発経験「あり」	443	68.6%
----------	-----	-------

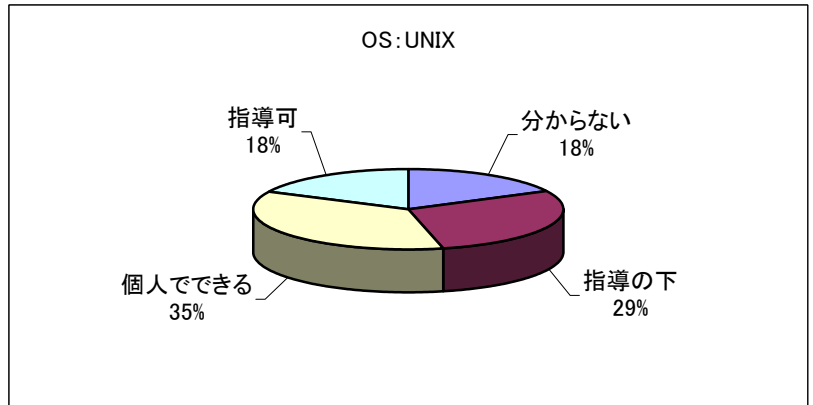
### 5-2-a-1. 開発プラットフォーム

プラットフォーム	数
パソコン	342
ワークステーション	142
メインフレーム	61
その他	13

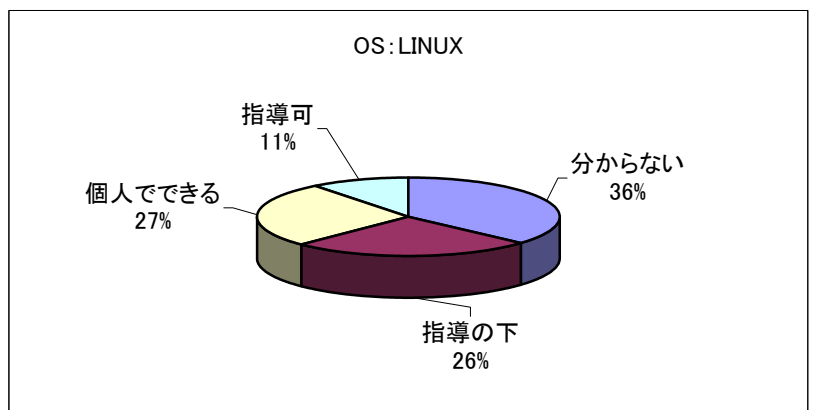


## 5-2-b-1. 習熟度:OS

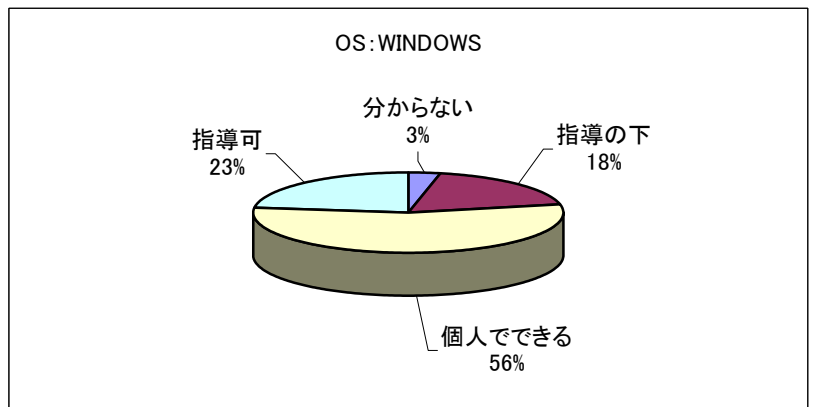
OS:UNIX	頻度
分からない	68
指導の下	112
個人でできる	139
指導可	69



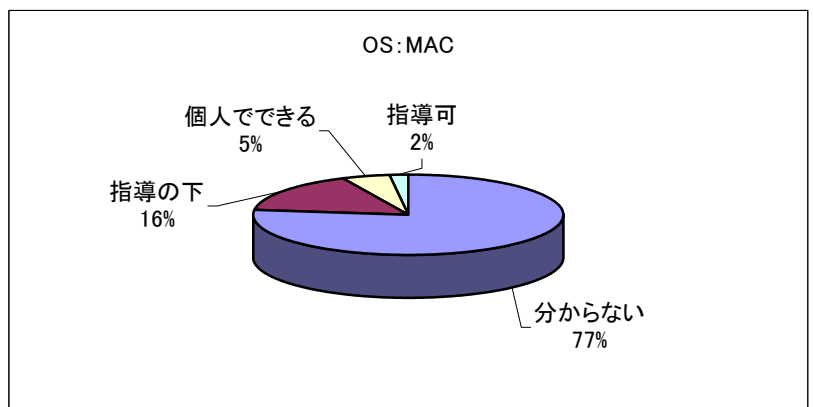
OS: LINUX	頻度
分からない	135
指導の下	94
個人でできる	99
指導可	39



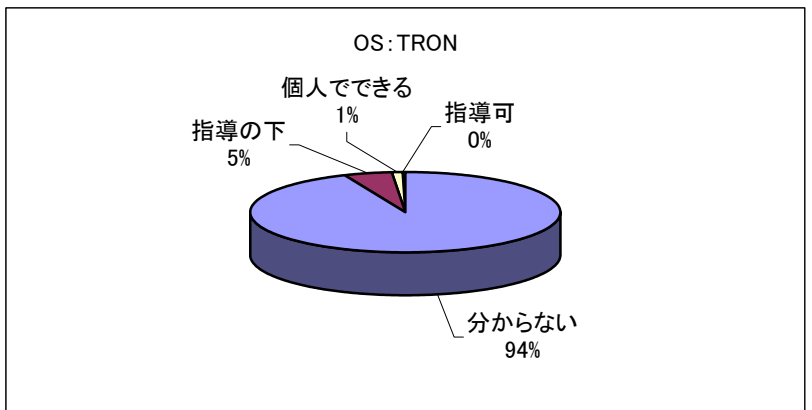
OS: WINDOWS	頻度
分からない	15
指導の下	82
個人でできる	249
指導可	103



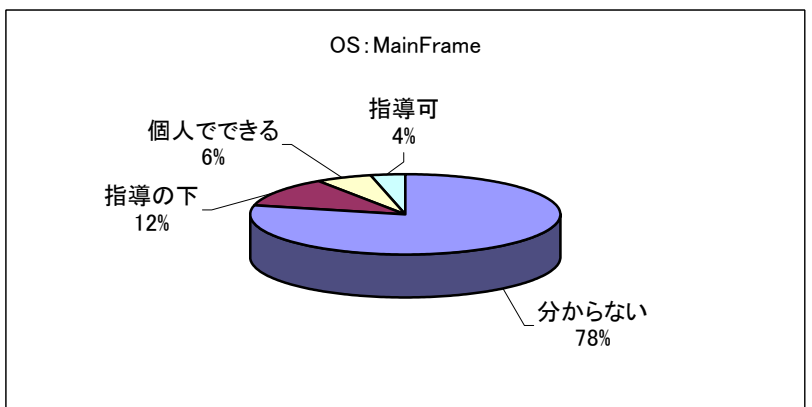
OS: MAC	頻度
分からない	271
指導の下	57
個人でできる	17
指導可	7



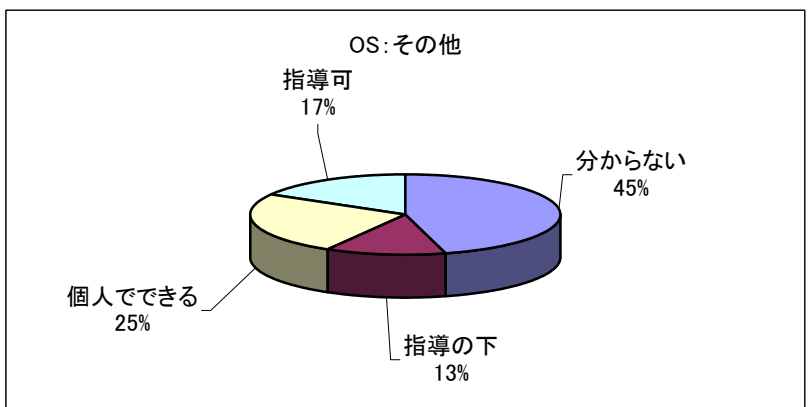
OS:TRON	頻度
分からない	330
指導の下	18
個人でできる	4
指導可	1



OS:MainFrame	頻度
分からない	288
指導の下	44
個人でできる	22
指導可	13

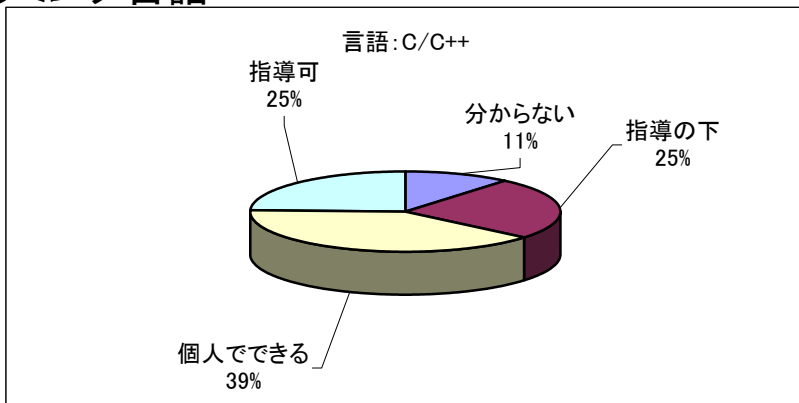


OS:その他	頻度
分からない	11
指導の下	3
個人でできる	6
指導可	4

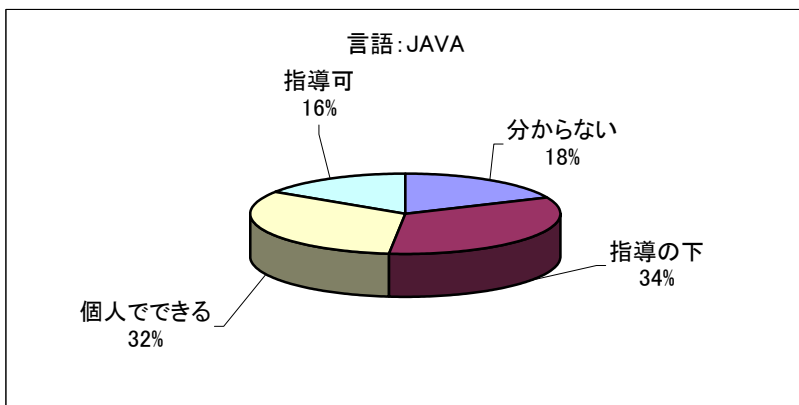


## 5-2-b-2. 習熟度:プログラミング言語

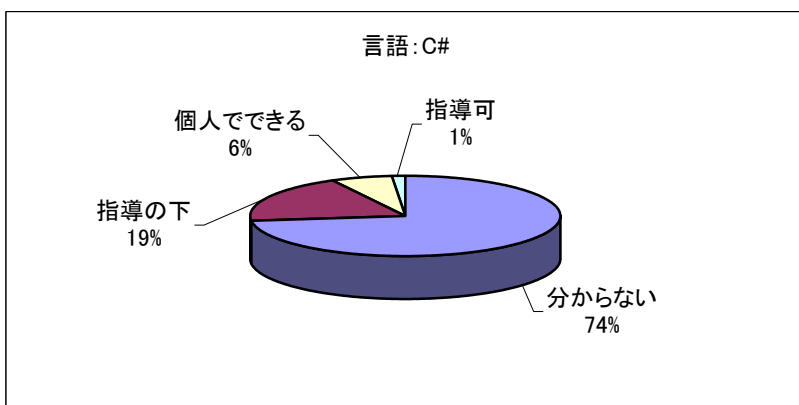
言語:C/C++	頻度
分からない	47
指導の下	105
個人でできる	165
指導可	103



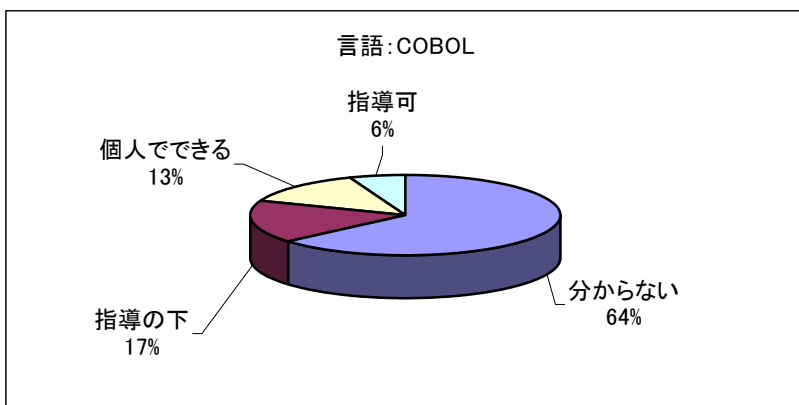
言語:JAVA	頻度
分からない	74
指導の下	134
個人でできる	130
指導可	64



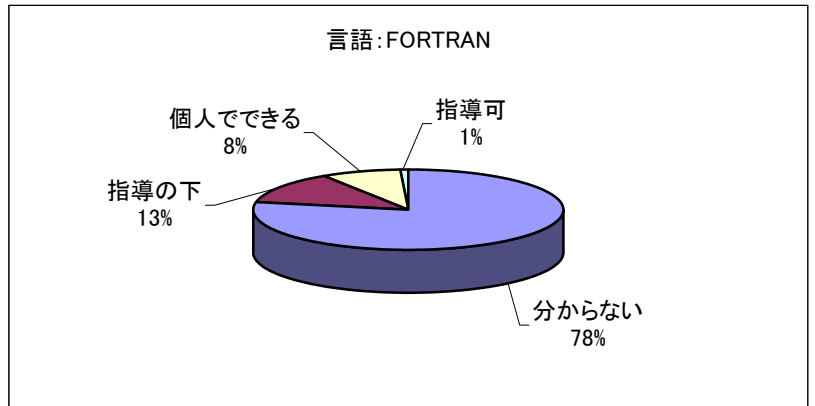
言語:C#	頻度
分からない	262
指導の下	69
個人でできる	23
指導可	5



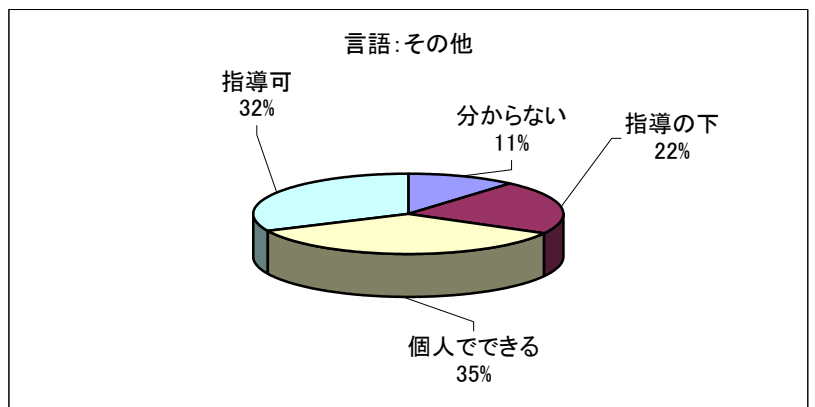
言語:COBOL	頻度
分からない	250
指導の下	68
個人でできる	52
指導可	23



言語:FORTRAN	頻度
分からない	277
指導の下	45
個人でできる	30
指導可	3

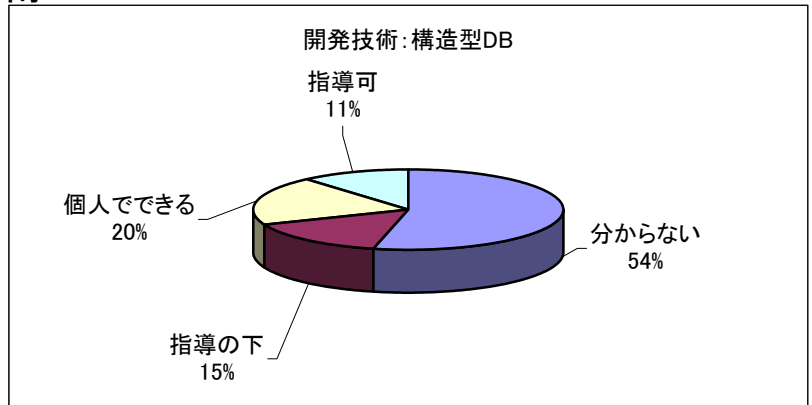


言語:その他	頻度
分からない	12
指導の下	23
個人でできる	37
指導可	34

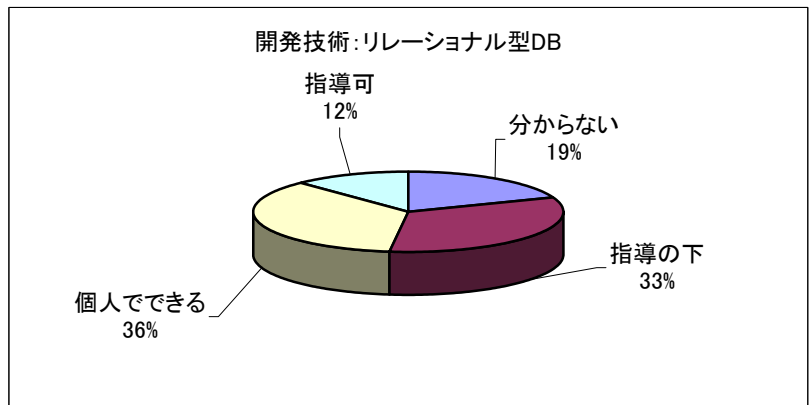


### 5-2-b-3. 習熟度:開発技術

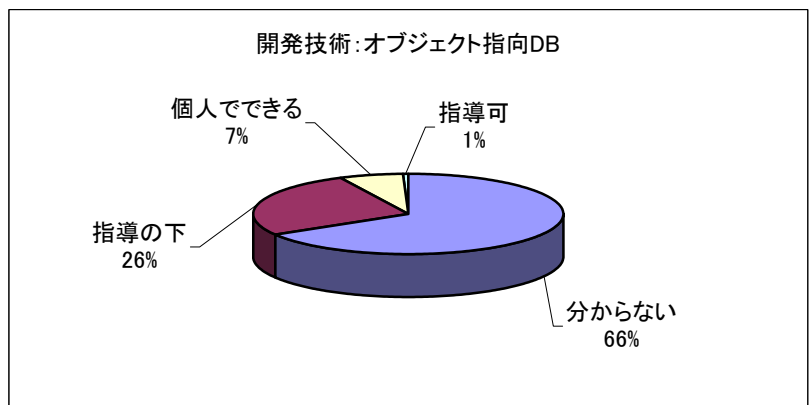
開発技術:構造型DB	頻度
分からない	38
指導の下	11
個人でできる	14
指導可	8



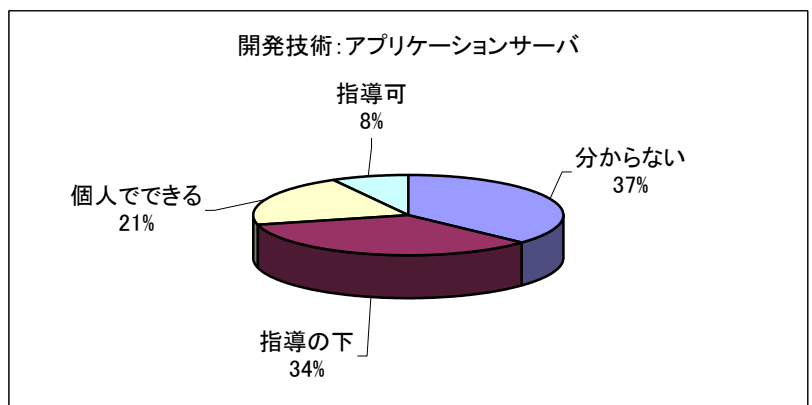
開発技術:リレーショナル型DB	頻度
分からない	79
指導の下	136
個人でできる	149
指導可	51



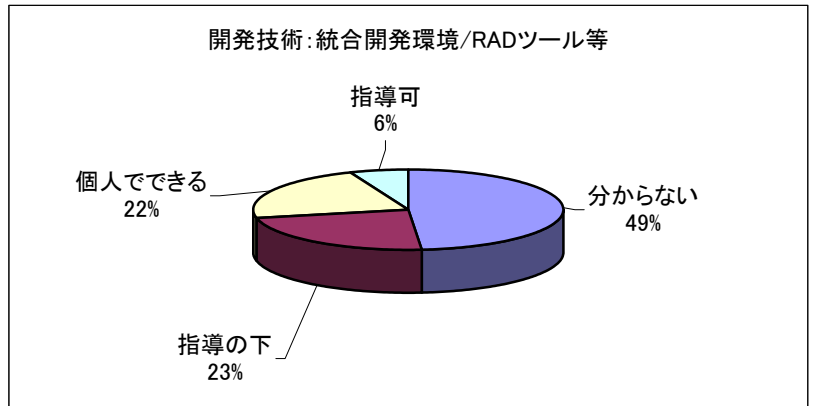
開発技術:オブジェクト指向DB	頻度
分からない	238
指導の下	94
個人でできる	24
指導可	2



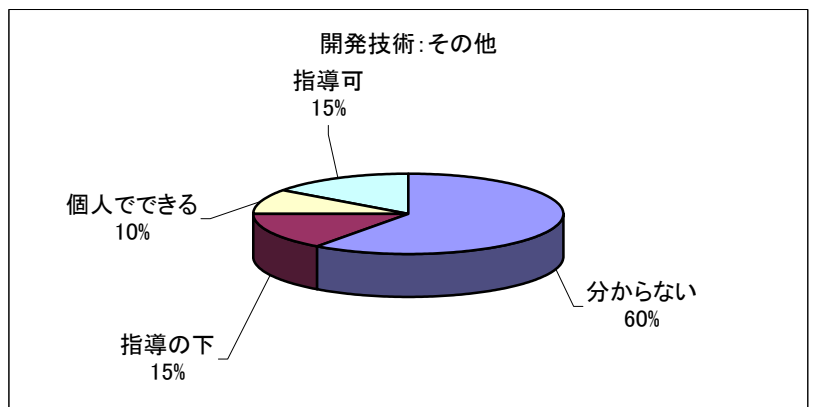
開発技術:アプリケーションサーバ	頻度
分からない	140
指導の下	129
個人でできる	79
指導可	31



開発技術:統合開発環境/RADツール等	頻度
分からない	177
指導の下	85
個人でできる	81
指導可	22

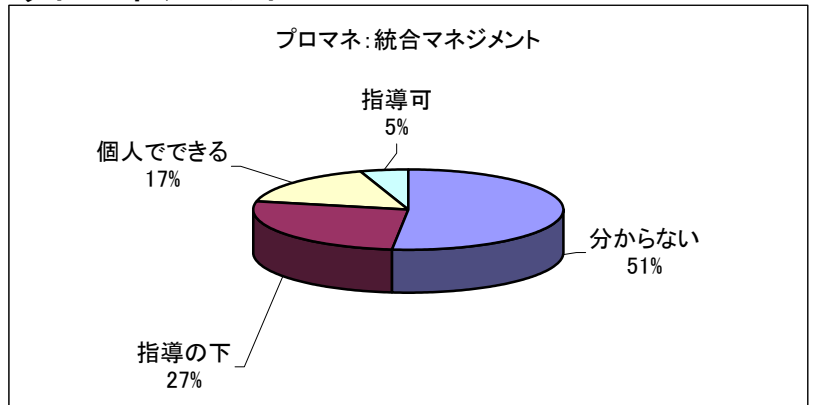


開発技術:その他	頻度
分からない	12
指導の下	3
個人でできる	2
指導可	3

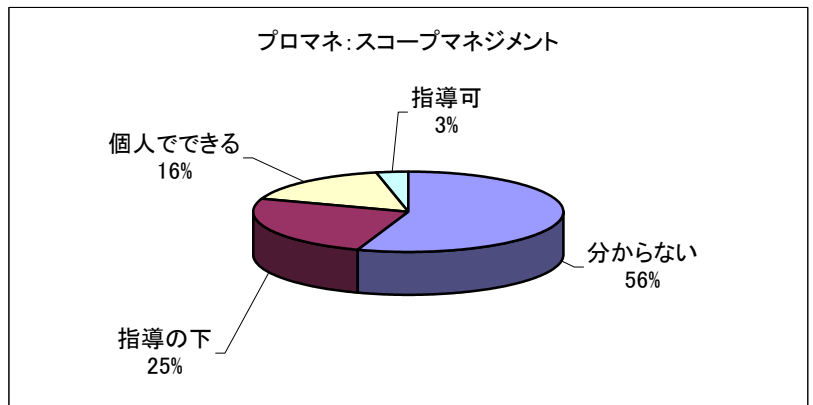


## 5-2-b-4. 習熟度:プロジェクトマネジメント

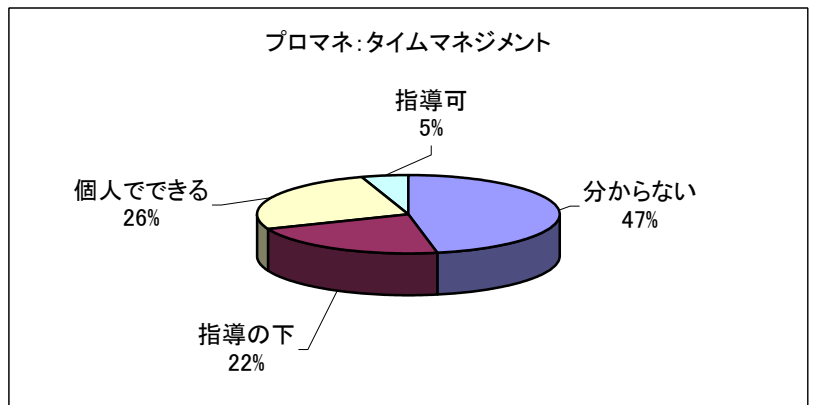
プロマネ:統合マネジメント	頻度
分からない	81
指導の下	42
個人でできる	26
指導可	8



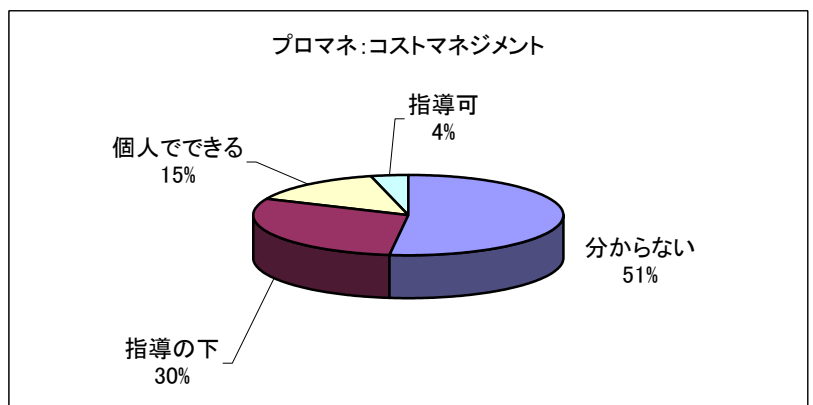
プロマネ:スコープマネジメント	頻度
分からない	84
指導の下	38
個人でできる	25
指導可	5



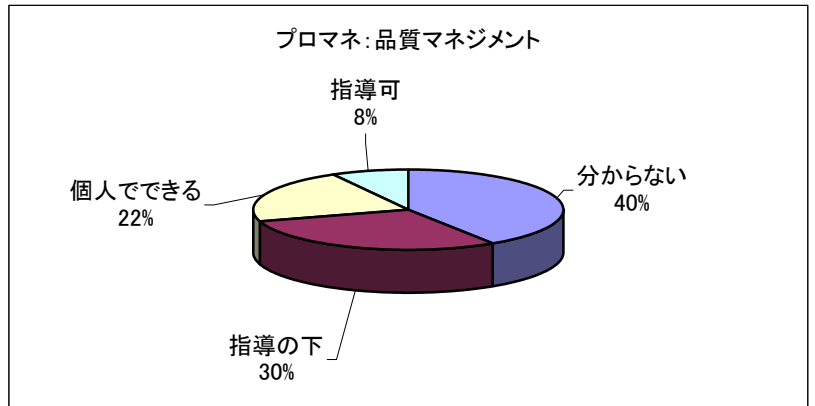
プロマネ:タイムマネジメント	頻度
分からない	75
指導の下	35
個人でできる	42
指導可	8



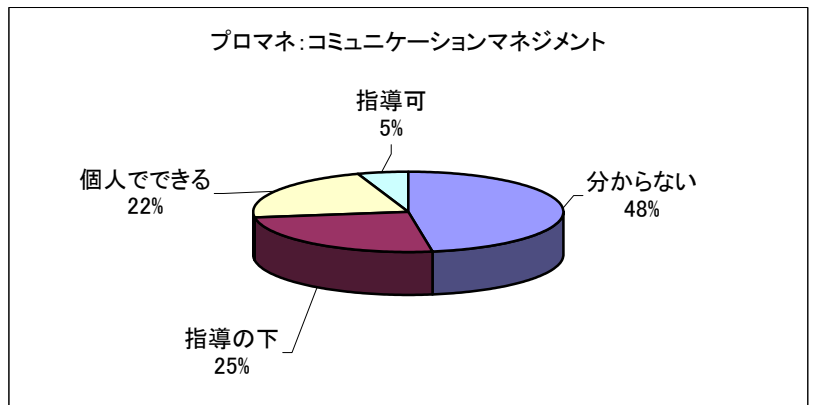
プロマネ:コストマネジメント	頻度
分からない	82
指導の下	47
個人でできる	23
指導可	6



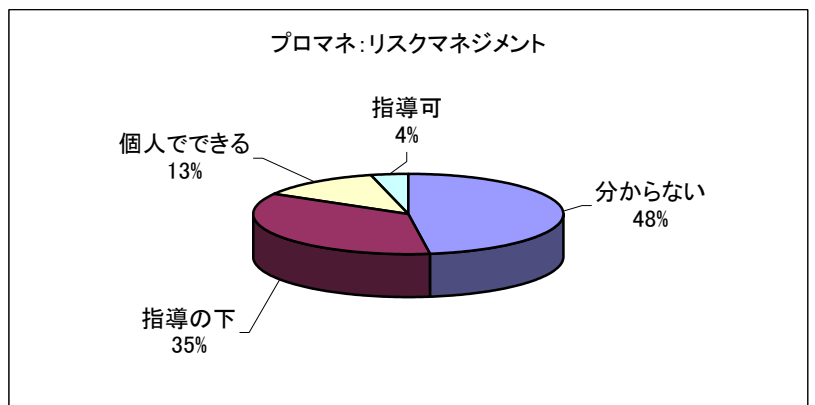
プロマネ:品質マネジメント	頻度
分からない	66
指導の下	48
個人でできる	35
指導可	13



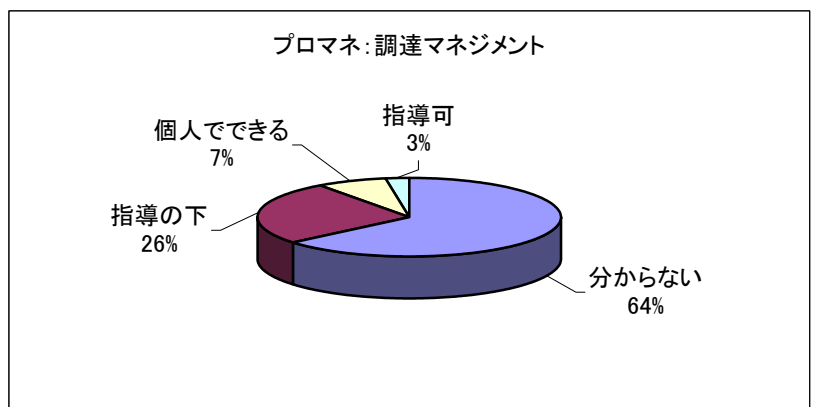
プロマネ:コミュニケーションマネジメント	頻度
分からない	73
指導の下	39
個人でできる	34
指導可	8



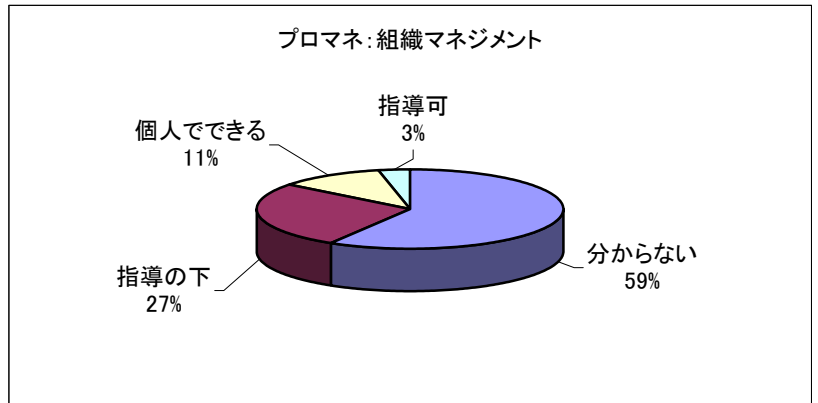
プロマネ:リスクマネジメント	頻度
分からない	74
指導の下	55
個人でできる	20
指導可	6



プロマネ:調達マネジメント	頻度
分からない	97
指導の下	40
個人でできる	11
指導可	4

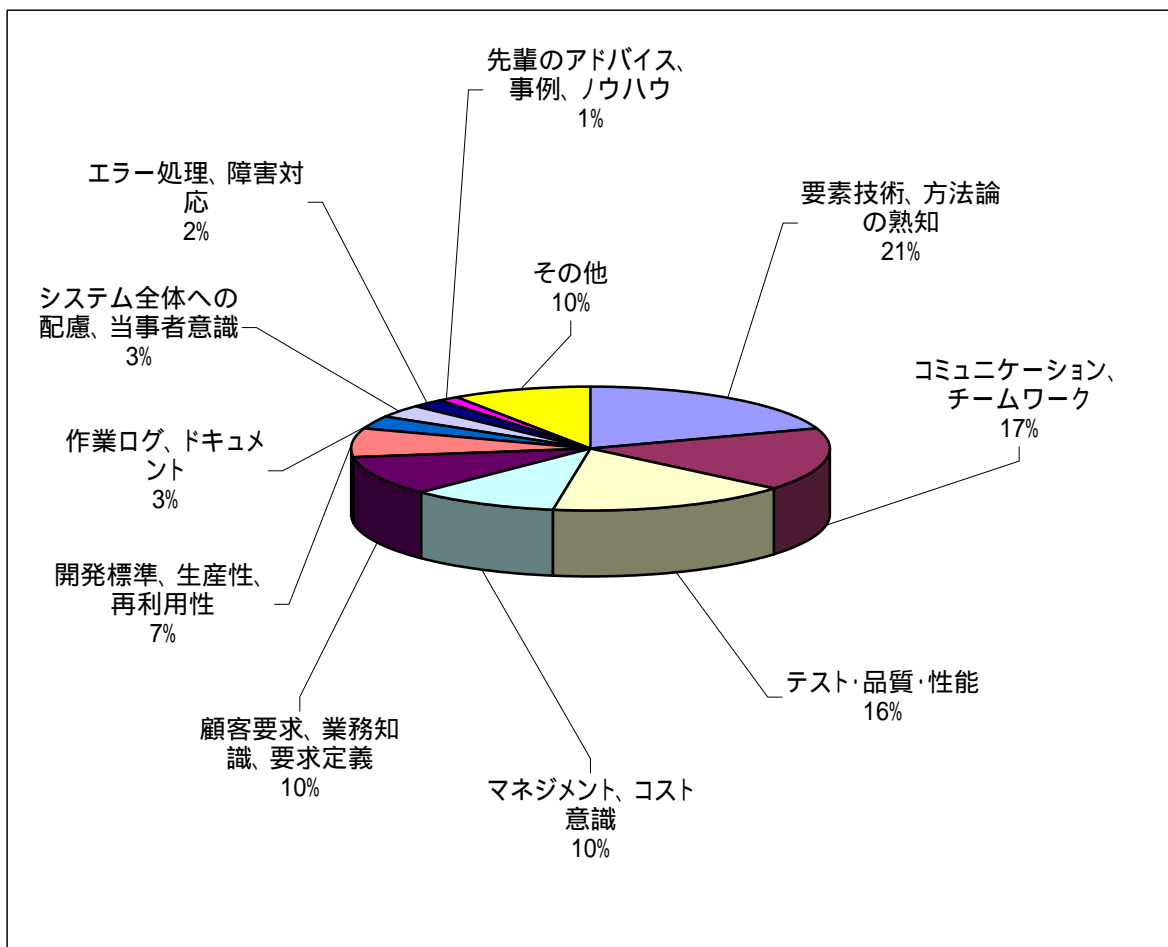


プロマネ:組織マネジメント	頻度
分からない	89
指導の下	41
個人でできる	17
指導可	5



## 5 - 2 - c - 1 . 開発時の特記事項の分類

特記事項の分類	頻度
要素技術、方法論の熟知	29
コミュニケーション、チームワーク	24
テスト・品質・性能	24
マネジメント、コスト意識	15
顧客要求、業務知識、要求定義	15
開発標準、生産性、再利用性	11
作業ログ、ドキュメント	5
システム全体への配慮、当事者意識	5
エラー処理、障害対応	3
先輩のアドバイス、事例、ノウハウ	2
その他	14



## 5-2-c-2. 開発時の特記事項(開発時に注意した点、勉強してた点等)

開発時特記事項
異国語でのコミュニケーション、カルチャー・ギャップを考慮して作業、会議を実施。時差というものが、重要な要因になることを気をつけました。
プロジェクトに参加している諸先輩方から多くのことを吸収するよう心がけた。
多人数開発での標準整備、品質と生産性の確保に注意した。
再利用性、可読性を考慮してコーディングする必要があること。
プログラムの再利用性に注意して開発した。
プロジェクト内の活発なコミュニケーションと、スキルに応じた仕事の配分に注意しました。
品質担保するために、どういったテストをすべきか。開発を進める上で、準備するものはどういったものがあるのか。
詳細な作業ログを残すこと。様々なソフトウェアを組み合わせ、最適なものをみつけること。
スケジュール管理、タスクの洗い出し、課題の把握
お客さんのことを考えて開発する。仕様を疑う。
テストを十分に行うこと。これに尽きる。
コスト・進捗・品質などを常に念頭において開発する点
開発者間のコミュニケーション
テスト手法、オブジェクト指向設計(デザインパターン)、パフォーマンスチューニング(APサーバ・データベース・OS)
フレームワークを利用したアプリケーション開発を行ったので、フレームワークの長所・短所を習熟し、他の案件にも生かせるよう心がけた。
新規開発ではなかったため、今までに行われてきた方法、流儀をいかに守り、統一性のある物にするかという点に注意した。
ソフトウェアテスト手法
他メンバーと協力しながら、チームとして最も高いパフォーマンスを実現すること。
自分自身が、無意識のうちに多くのことを考慮して開発を行っていることに気付いた。これにより、他人を指導するときに、何を教えれば良いかが分かるようになった。
ユーザーにとって分かり易いインターフェースを心がけた
色々ありすぎてもう…。限界です。
注意した点:対向ノードとのインターフェース設計。テスト時のトラブル解析において、無駄なく迅速に対処すること 勉強した点:トラブル発生時に、障害原因の改修をどの部内で行うかの調整等
能力の低い人を、すばやく発見する方法
仕様変更への柔軟な対応
全体の進捗が遅れないように注意した。
SQLは、独学で習得
プロジェクトマネジメント
プラットフォーム(OS間)での実装、内部処理の違い
1人の考えだけで作業を進めず、周囲との調和をはかること。
プログラムを一面作るのに、様々な観点でテストを行う点。コーディングする時に、他の人にも分かるように、分かり易いコメントを追加した等。
小さなシステムですが、基本設計からテスト・リリースまで全てやっているので、システム開発の流れを身をもって体験しました。また、ドキュメントの履歴管理・懸案管理を学びました。メンバーとの解釈の違いをなくす努力も大事だと思いました。

開発時特記事項
ネットワークプロトコル及びプロトコルの実装について勉強した
開発の上流工程で、十分な検討を行うこと。
基本をしっかり学ぶこと。 関連する知識を常に学ぶこと。
仕様が曖昧な場合には、案をいくつか提示し、発注側に選択してもらった。
システム全体の動き及び業務の内容を、ある程度理解した上で取り組まないと、自分の開発について自信が持てないし、疑問に感じる事も何もない。内容の理解を十分に行うようにした。
テスト等を行う際に、その試験環境や試験そのものを他の面罵が簡単に行えるように手順書等を作成して残しておくという点
性能及びメンテナンス性に注意している
業務(銀行業務)を意識、勉強して開発
visual basic がよく使われており、ツール類も多種存在する。 Microsoft関連の知識も必須
チームの情報共有、ナレッジ管理の重要性
ソースコードのリファクタリング。テスト仕様の詳細を先にけっいていすること。開発の前にテスト環境ありき
クリティカルシステムに特化した考え方(リスクマネジメント)
チームで作ったものに対して必ず第三者的に人に再度確認してもらう
各OSに依存したコンパイラの仕様の違い。各OSのアーキテクチャの違い、すべてのエラーケースを洗い出す作業、限界値の考慮などは開発をするようになって始めて知った。
グループを組んでみてはじめて人は能力を発揮します。ということがよくわかる
開発言語がCOBOLということもあり、結構古い人材が多く、縦の関係で、自分の意見がいえない部分があり、最終的に大きな障害が起きたりした。
MPEGやDVDの規格について必要部分を調べた
システム変更時の影響範囲を把握することが重要。システム変更内容だけでなく、そうする理由を具体的に提示することが重要。
システム開発の規模見積もり方法を意識して取り組んだ
チームワーク、マネジメントを理解する
ハードウェアの知識
設計の段階で書類を作成するのに苦労した。他の人が見ても開発を進められるよう、自分には理解できないものにならないよう注意した。
効率化、作業時間(コスト)の削減をこころがける。誰が見ても理解できるプログラム。他に使えそうなところは汎用的に作成する。
上流工程で品質はほとんど決まる
研究業務なのでシステム開発とは少しちがいます
バグを作りこまないことを第一に考えた
開発においては必要に応じて、また独自の勉強には調べてもどうしてもわからない場合は上司や先輩に聞く(すぐには頼らない)
PMBOK等の一般的マネジメント技法
ドキュメントを充実させる点に注意した。
テスト技法、バグ分析
スケジュール管理、ミーティングの重要性

開発時特記事項
新しい技術を使うプロジェクトに対して、会社が教育してくれることはほぼ無い。その為、自ら学ばなければならない。私の場合も初めてのJava開発がEJBであった為、非常に苦労した。
レビューの大切さ、仕様の時点で不具合を発見することで工数が大幅に減る。
注意した点・・・フォルダの構成、ファイルの更新、コーディング基準など
プロジェクトメンバーに指針を示す際に、理解しやすく、形に残していくこと。
工程全てが新たな勉強。
・情報を収集し、理解する能力の必要性を感じた。 ・読解能力が重要だと思う。
オブジェクト指向とDB
仕様、業務の理解
規模が大きくなったときや、高負荷となったときも考りよして設計する
品質管理には十分に気をつかいます。
異常系の重要性。正常系はもちろんだが、異常系の対応も必要。
報告、連絡、相談を新人なの特に注意しました。
DBの正規化、入力値チェック、セキュリティ
人に見やすいコーディング
CORBA技術を適用したプログラミングの開発。C++によるマルチスレッド対応オブジェクト指向プログラミング。マルチプラットフォーム対応のソースコードを作成
いかに仕様を満たす信頼性の高く保守しやすいコードを書くか、そのためにそのように設計するか。要求仕様分析等々
お客の話聞く。
VBに触れた事がなかったので、VBの勉強をしました。後は、自分の勝手な判断で進めずに、確認しながら進めました。
品質管理に特に注意。バグを作りこまない等
リレーショナルDBの仕組みなどを学んだ
設計時にしっかりとしたアルゴリズムを作成すること。
品質
ソフトウェア開発時は、正常系と同様に異も重視してコーディングおよび評価を行うこと。
オブジェクト指向
実際のプログラミング・テスト作業に対して品質向上の重要性を理解させること。そのために、エラー処理などで手を抜かないコーディングと十分なテストが必要であることを待たせざるに苦労している(「動けばいい」という考え方ではシステムは開発できないということ)
品質確保の重要性を学んだ。
品質を重視、納期内でできればよいのではなく、納期内にかつ品質がよくなくてはいけない
プロジェクトの進捗管理と品質管理
プラットフォームの意識、エラー処理方式の意識、ユーザーインターフェース、使用するライブラリ、アプリケーションのサイズ
・学生時代の演習では、簡単な入出力を持った関数レベルしか学んでいないため勉強しなければならない点が多かった。 ・GUIの開発の経験もゼロ ・プラットフォーム毎の違いに関する意識も当初はなかった。
品質を向上させるように開発する。バグの作り込みを防ぐよう開発する。

開発時特記事項

開発言語の勉強をしたりした。また、仕様に基づき、その仕様が実現可能であるのかの判断等に注意して、開発にと取り組んでいる。
納期と品質、品質に関してはテスト技法は従来の技法を用いるが設計、開発においては個々の能力に任せるよう心がけている。(担当にスキルがある場合は)
別の人にもわかりやすく。
プログラム構造、処理について勉強しました。
OJT中なので作業の流れに注意をした
既存のシステムやプログラムを見習うことが大事だと感じた。日常的に知識を累積するのが大事だと感じた
初期は主に改造が多かった為、処理の流れなどを理解しながら、システムの動作と結びつけながら作業するようにした
開発の行い方、評価方法等を勉強
開発標準を厳守することが、後のトラブルを未然に防ぐ近道。
技術はもちろんだが、進捗管理や資料の作成法等、実務以外の場ではなかなか学びづらい事を身につけるよう注意している。
報告、連絡、相談の重要性、特に報告
ここ最近では、PMBOKに準拠し、作業を行うことにしている。
業務知識の習得
担当者間の認識を合わせる。納期を守るため、定期的に進捗報告(会議)を行う。仕様変更に対応できる仕組み(設計時)づくり。
コミュニケーション能力の必要性。ドキュメント作成、精微化の重要性。
保守性と運用性
独学が基本。
設計、プログラミング、テストの各工程において、設計書・仕様書・テストチェックシート等のドキュメントを作成し、できればユーザーを交えてレビューを行う。
品質、コスト、納期を両立させること。
全体での意識合わせ、意思交換
MS-WINDOWS-NT、95で、業務タスクと通信手順ライブラリ(BSC、全銀)の間で動くサブタスク(ミドルウェア)を社内独自開発し、業務アプリの通信制御の手間を軽減した。(タスク間通信、同期、スレッド等、システムプログラミング的な知識、経験が役に立ったと思う)
設計に力をいれ、仕様を固めることが後の工程で重要となる。
大規模開発のため、他グループとの連絡を密にし、後戻り工数が発生しないようにする。
基本的に複数人での開発になるので、コミュニケーションを良くとるようにしないとイケない。コミュニケーション不足による不良は後のテスト工程で出てくる事が多いからである。
開発の現場で要求分析からテストまでの一連の流れを学んだ。
仕様書はしっかり読もう
開発を行う前1日ごとスケジュールを立てること。ものごとに対して根拠をもつこと。
しっかりとした基本設計書を作成すること。
仕様とユーザーニーズの相違点に毎回苦労させられているので、早い段階でユーザーを巻き込んでネゴを細かく取るようにしている。
ドキュメントの書き方、品質の大切さ
テストの効率化と確実性の向上

開発時特記事項
他の人が分かりやすいようなコメント文や説明文になるように注意した。
リスクを考えたスケジュールを考えておくこと
知識よりもコミュニケーション能力の方が重要だったと痛感。自分達の常識をかみくだいて顧客に伝える能力が何よりも重要。
懸案事項、注意事項等のメンバー間の情報共有と共通認識
スケジュール調整、課題管理
開発には参加したが実際に開発したわけではなく、PJ内の雑用をまかされていた為、技術的な習得はできなかった。
オブジェクト指向に基いたシステム開発
スキル、レベルの違うメンバーを理解するのは重要だと思った。
プロジェクトマネジメントとのかねあい
システム内外の計画化、パフォーマンス、リリース問題、エラー処理、スリーラビリティ、保守性
複数の人員で開発するため、連携・コミュニケーションには気を配った。開発工程手順は実務をもとにした書籍に目を通すようにした。
様々、OJTに勝る経験は無いと思う。
数値計算での誤差逓減、計算高速化
お客様とのコミュニケーション
他人に見やすく、分かり易く、効率良く。
・品質向上を行う手順(チェック観点・テスト方法) ・多人数開発における、資源管理(内容変更の際の構成管理手順も含む)
・品質に関しては、得にパフォーマンスを重視した ・プロジェクトに参画することで、曲面ごとの体制のあり方や、人間系の問題に関する経験を身につけた
オブジェクト指向、ソフトウェア工学(開発プロセス・テスト技法)
周囲とのコミュニケーション。忍耐力。SEにとって、最も重要なのはコミュニケーション能力であるということ。
コミュニケーション(設計者・開発者の認識の統一) 色々なソースを見て、手法を学んだ。
効率
痛感したのは、1ステップの修正でもかなりシビアである。趣味のコーディングをかけ離れて、実際の会社でのソフト開発は相当集中力が必要。その分、ストレスがかなりたまる。
他の人が見ても分かり易くプログラムを書く。

## 6. 現在の業務について

6-1-a-1. 現在の業務の経験年数

6-2-a-1. 一人当たりの回答業務数

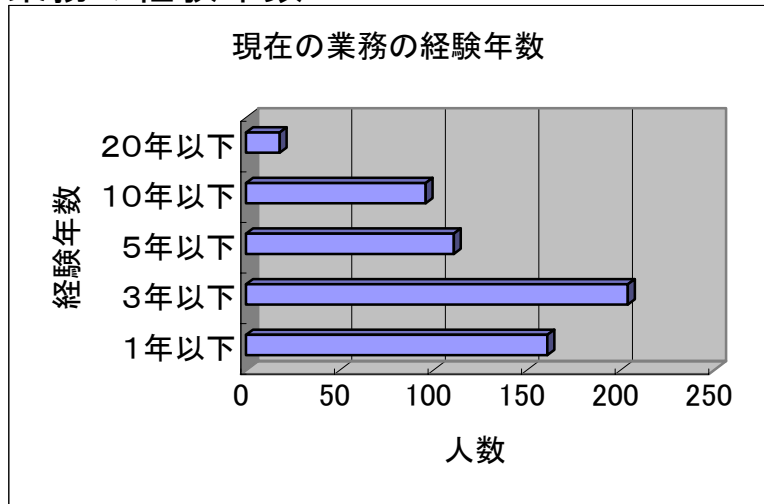
6-2-b-1. 業務分類毎の回答件数

6-2-b-2. 業務分類毎の経験年数構成

6-2-b-3. 業務毎の回答件数詳細

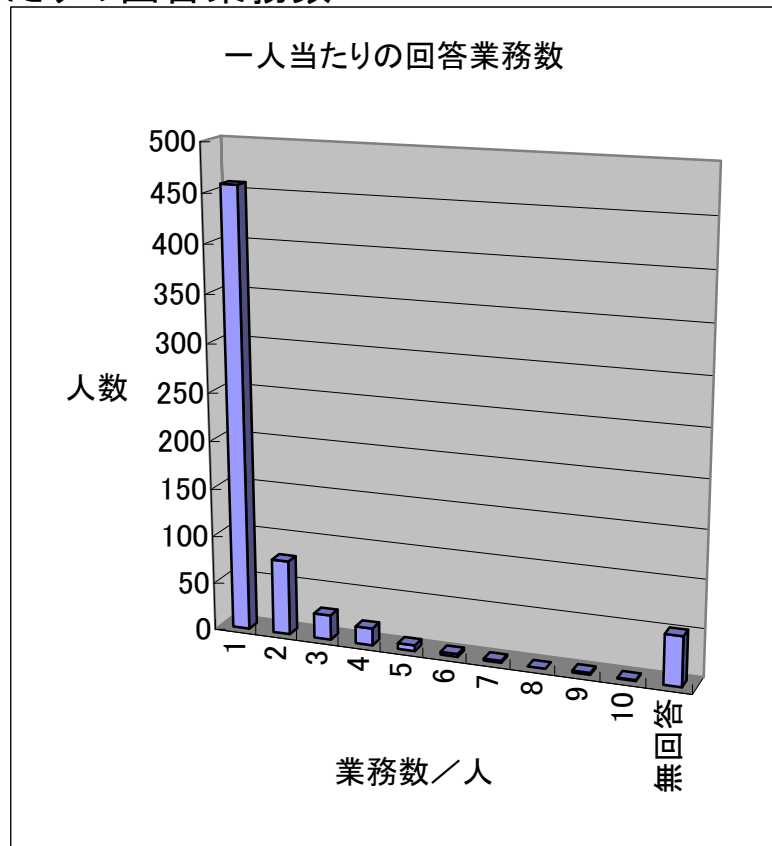
### 6-1-a-1. 現在の業務の経験年数

経験年数	人数	比率
1年以下	161	24.9%
3年以下	204	31.6%
5年以下	111	17.2%
10年以下	96	14.9%
20年以下	18	2.8%



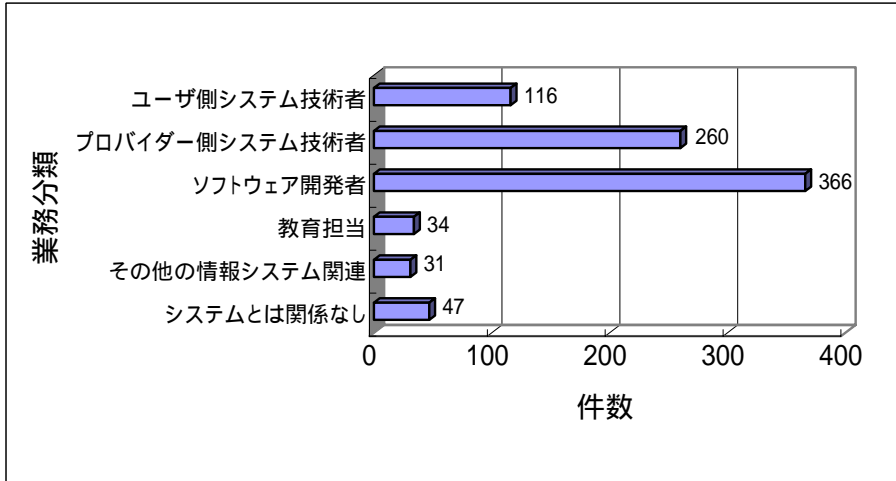
### 6-2-a-1. 一人当たりの回答業務数

職種数/人	人数	比率
1	458	70.9%
2	78	12.1%
3	26	4.0%
4	18	2.8%
5	6	0.9%
6	3	0.5%
7	2	0.3%
8	0	0.0%
9	2	0.3%
10	1	0.2%
無回答	52	8.0%
合計	646	100.0%

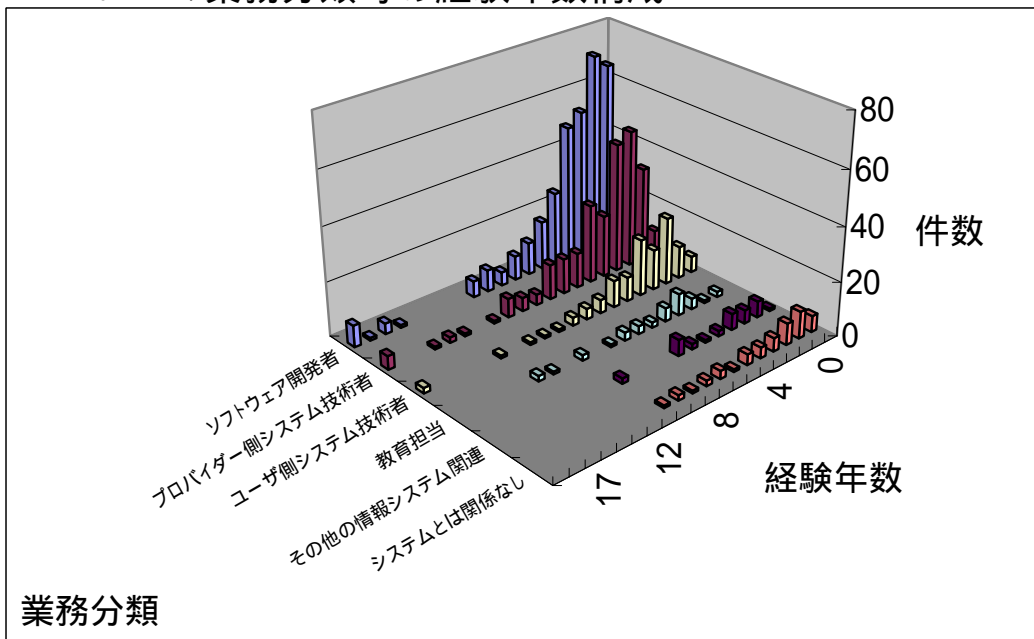


### 6 - 2 - b - 1 . 業務分類毎の回答件数

職種分類	件数
ユーザ側システム技術者	116
プロバイダー側システム技術者	260
ソフトウェア開発者	366
教育担当	34
その他の情報システム関連	31
システムとは関係なし	47
合計	854



### 6 - 2 - b - 2 . 業務分類毎の経験年数構成



6 - 2 - b - 3 . 業務毎の回答件数詳細

職種分類	職種	件数	小計	合計
ユーザ側システム技術者	プロバイダーへシステム要求	11	116	
	ユーザ側システムの運用保守	85		
	ユーザ側でのシステム指導者	18		
	その他	1		
	ストレージ導入サービス	1		
	商品説明	1		
プロバイダー側システム技術者	システムインテグレーター	92	260	
	プロジェクトマネージャー	27		
	開発者(アプリケーション)	125		
	開発ツール開発	16		
ソフトウェア開発者	企画・計画・マーケティング	22	366	
	研究・開発	234		
	テスト・QA	79		
	サポート・販売支援	31		
教育担当	社内教育	25	34	
	社外教育	7		
	その他	1		
	パソコンよらず相談所	1		
	(空白)	1		
その他の情報システム関連	システム関連コンサルタント	24	31	
	エバンジェリスト	3		
	その他	1		
	カスタマーサポート	1		
	キャリア開発	1		
	プロジェクトリスクコントロール	1		
	(空白)	1		
システムとは関係なし	HDLによるLSI設計	1	47	854
	LSI設計	1		
	コンサルタント	1		
	セキュリティ技術者	1		
	デジタル放送標準化、規格策定、受信端末	1		
	デバイス、ハードウェア研究開発	1		
	ネットワーク運用・保守	1		
	ネットワーク保守、運用	1		
	パッケージアプリケーションソフトウェアの	1		
	ビジネスコンサルタント	1		
	運用・保守	1		
	営業	11		
	営業・企画	1		
	開発によるデータ管理	1		
	技術系公務員(電気)	1		
	業務改革、過去にシステム開発	1		
	金融業	1		
	経営者	1		
	研修	1		
	光通信関連	1		
	公務員	1		
	高周波図 技術	1		
	高等専門学校教官(および大学院)	1		
	自動車用電子システムの使用設計	1		
	人事	2		
	総務、経営管理	1		
	地方公務員(電気職)	1		
	調査・研究	1		
	通信ネットワーク保守	1		
	通信関係(キャリア)	1		
	電気関係の研究(必要なソフトはプログラミ	1		
	土木行政	1		
	販促、法務	1		
	放送局の技術システムにおけるネットワーク	1		
	無職	1		
	理科教師	1		

## 7. 現在、研修または自分で勉強していること

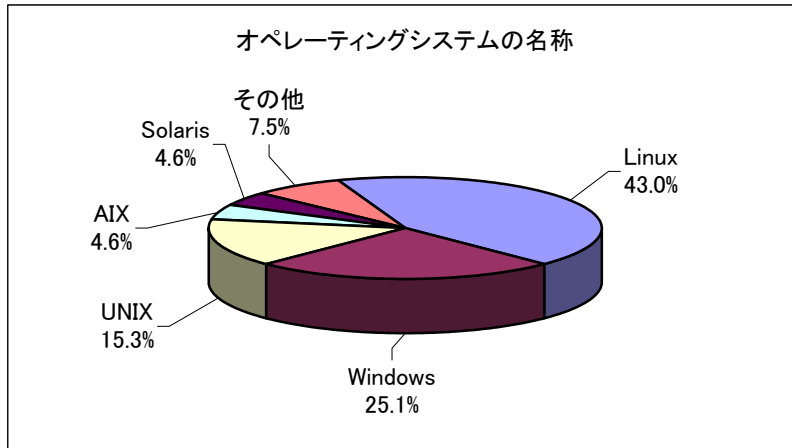
- 7-a-1. オペレーティングシステム
- 7-a-2. 言語
- 7-a-3. データベース
- 7-a-4. 開発ツール
- 7-a-5. ネットワーク／セキュリティ
- 7-a-6. その他

## 7-a-1. オペレーティングシステム

オペレーティングシステムの名称(複数回答) 件数

Linux	132
Windows	77
UNIX	47
AIX	14
Solaris	14
その他	23

回答者
250

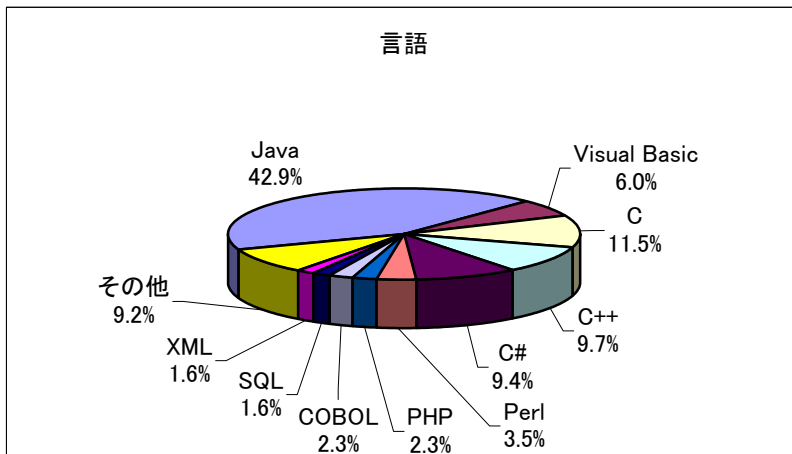


その他回答例
Fedora Corel
FreeBSD
HP-UX
Mac
OpenVMS
OS/400
SNMP
T-Kernel
TRON
Visual Studio .NET
z/OS, z/Linux
メインフレーム
リアルタイムOS数種

## 7-a-2. 言語

言語(複数回答)	件数
Java	186
Visual Basic	26
C	50
C++	42
C#	41
Perl	15
PHP	10
COBOL	10
SQL	7
XML	7
その他	40

回答者
316

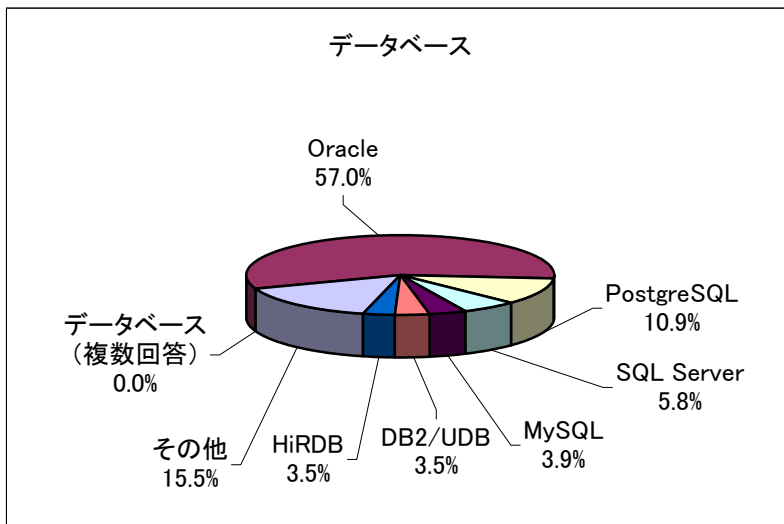


その他回答例	
ASP	PL/SQL
ant	python
awk	Ruby
BML	SAS
C3	Scheme
CURL	Servlet
ECMASクリプト	Sh
EJB	Shell script
HTML	SQL
iXML	Vbscript
JavaScript	Web作成用言語(PHP, CSS)
Java1.5, C#	アセンブラ
JSP	アセンブラ(intelプロテクトモード)
Lisp	サーバサイドJavaSQL
Pascal	各種スクリプト言語
PL/I	

### 7-a-3. データベース

データベース(複数回答)	件数
Oracle	147
PostgreSQL	28
SQL Server	15
MySQL	10
DB2/UDB	9
HiRDB	9
その他	40

回答者
232

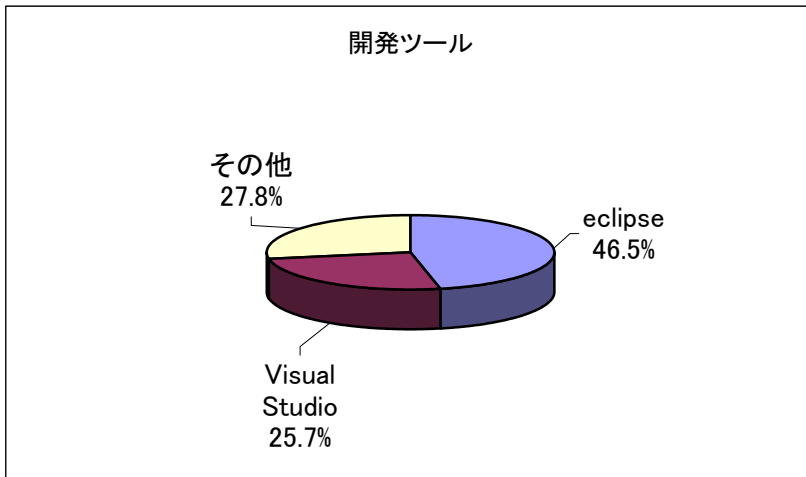


その他回答例
Access
SQL
InterBase
Java
MSDE
NonStopServer
NXDB
PL/SQL
RDB一般
Saybase
XML
オブジェクト指向DB

## 7-a-4. 開発ツール

開発ツール(複数回答)	件数
eclipse	67
Visual Studio	37
その他	40

回答者
134

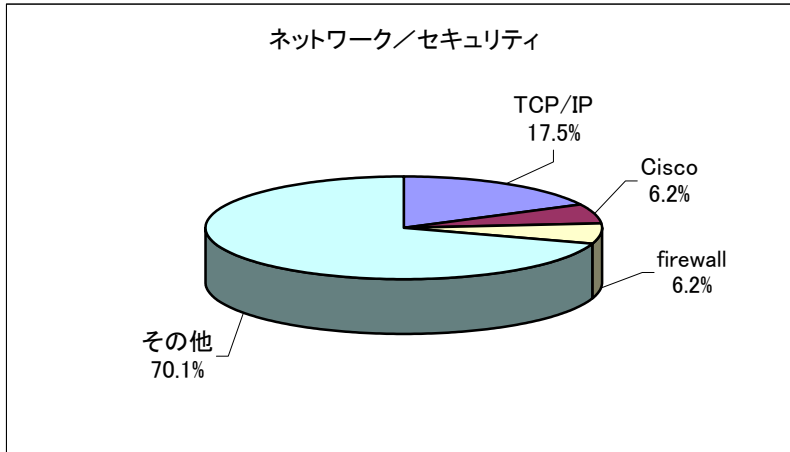


その他回答例	
2end studio	J-Rat
Access	Jude
Advantage Gen	Orteus
Ant	Rose
Blender	Struts
Cosminexus	TOMCAT
CVS	vi
Cygwin	Web Matrix
Developer	Weblogic
Emaes	WebLogic Workshop
Enhydraアプリケーション	Webサービス関連
gee	Windows DDK
GNU	WSAD
GNV(GCC)	ZOPE
IBM/Rational系ツール	スクリプト
Java	テキストエディタ
JBuilder	モデリングツール

## 7-a-5. ネットワーク/セキュリティ

ネットワーク/セキュリティ(複数回答)	件数
TCP/IP	17
Cisco	6
firewall	6
その他	68

回答者
89

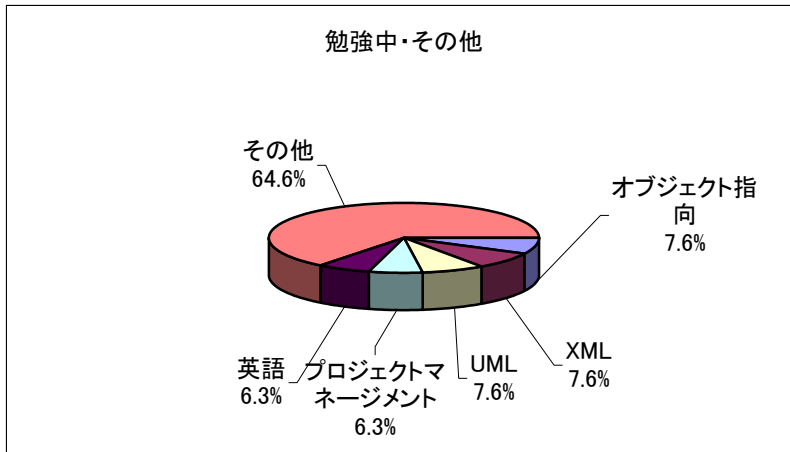


その他回答例	
NET Framework コードアク	selinux
Alteon	SNMP TCP/IP
CCNA	SOAP
CORBA	Socht programing.
DES	SoftEcler
Fibre Channel	SSL
HTTP	VPN
HTTPサーバ管理	WEB
IDS	Web-Service
IP8800/700	クラスタ
Ipub	システム基盤構築
IPV6	システム設計・運用
IP-VPN	セキュリティ
ISMS	ネットワーク監視製品
LAN管理	ネットワーク基礎
Layer3 switch	ネットワーク攻撃への対策
LDAP	ネットワーク全般
NAT	メールサーバ管理
NET Webサービス	ルーティング
NetFilter iptables)	基礎
NW監視	除法処理試験
PKI	情報処理技術者(ネットワーク)
PLX	電子証明書
Real Secure	無線LAN
RTP/RTCP	

## 7-a-6. その他

勉強中・その他(複数回答)	件数
オブジェクト指向	6
XML	6
UML	6
プロジェクトマネジメント	5
英語	5
その他	51

回答者
82



その他回答例	
Apache	ソフトウェアアーキテクチャ
BGP	ソフトウェアパターン
CG検定一級の勉強	ソフトウェア開発手法:XP
CIM	データマイニング
CMS	デザインパターン
HP製品のOpen View	ドキュメンテーション技術
iscsi	ハードウェア知識
J2EE	プログラミングテスト技法
JSP	プログラムマネジメント
MPLSオペレーティング	プロジェクト開発手法
NtoN TV会議	プロセス改善
Open/Tp1	ユーザビリティ工学
PHP	業務知識の習得
SAN	経営戦略
SAP R/3	自動テスト手法
SCSI技術	自律コンピューティング
Tivoli	出先、自社、自宅どこからでも使える文房具
Tomcat	情報数学
VERITAS	色彩検定
Vmware ESX/GSX/Works	心理実験
WDCフレームワーク	総合的なプロジェクトマネジメントの実践
Webサービス	電気工学
インストラクションデザイン	統計学
クラスターサーバ技術	統計的分析手法
サーバ構築について	簿記
サーブレット	要求工学
システムアーキテクト	量子情報科学
システム開発の工学的な知識の習得	量子情報理論

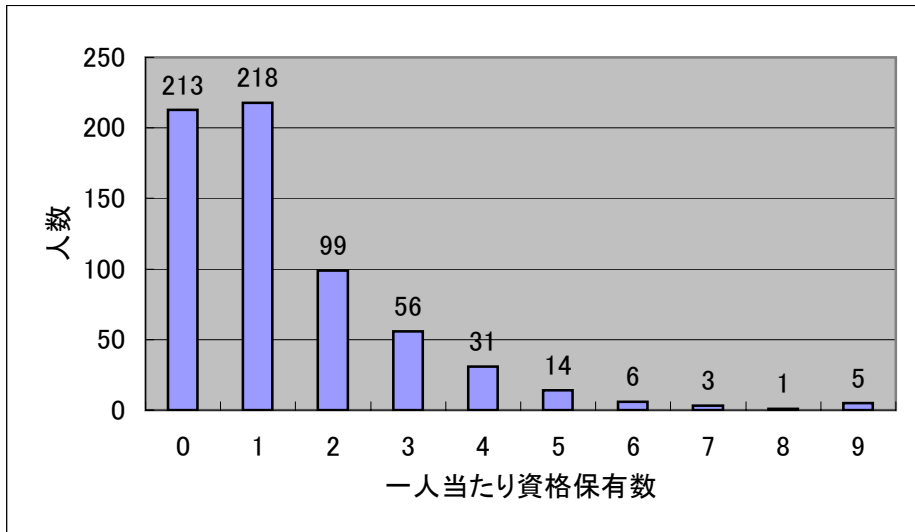
## 8. 資格(取得済み・目指している資格)

- 8-a-1. 資格の平均数
- 8-b-1. 取得済みの資格数
- 8-b-2. 目指している資格数
- 8-c-1 種類毎の資格保有数

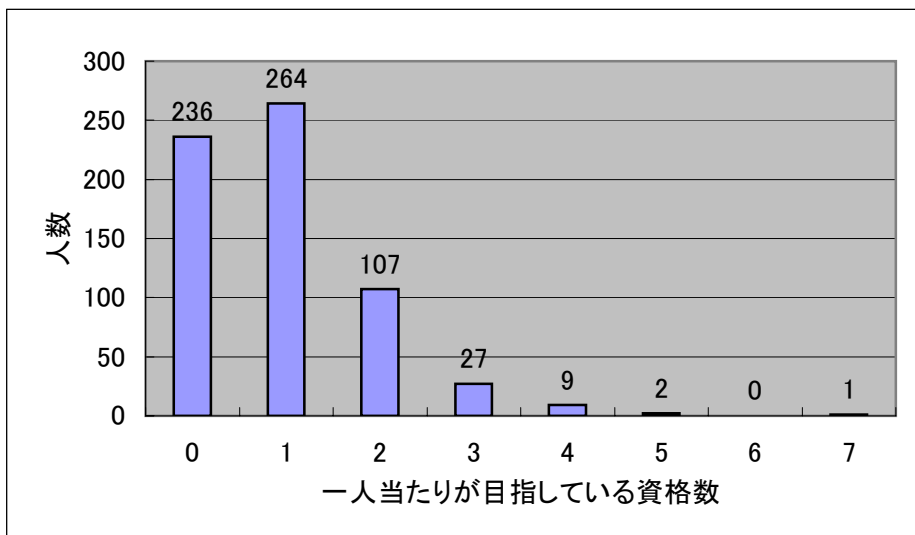
### 8-a-1. 資格の平均数

保有資格平均数	1.38
目指している資格平均数	0.95

### 8-b-1. 取得済みの資格数

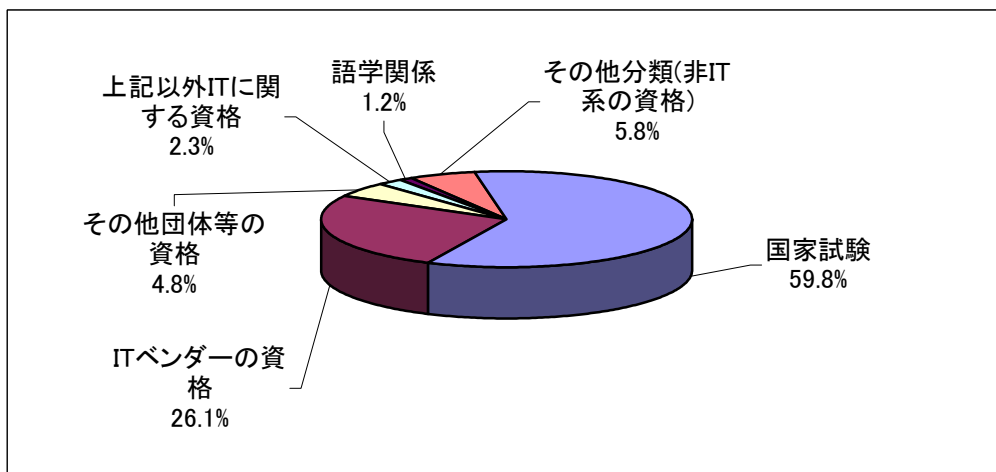


### 8-b-2. 目指している資格数



### 8-c-1. 種類毎の資格保有数

<b>国家試験</b>	<b>540</b>
基本情報技術者試験	284
ソフトウェア開発技術者試験	134
初級システムアドミニストレータ	36
情報セキュリティアドミニストレータ試験	15
アプリケーションエンジニア試験	11
プロジェクトマネージャ試験	4
テクニカルエンジニア(ネットワーク)	40
テクニカルエンジニア(データベース)	12
テクニカルエンジニア(システム管理)	3
テクニカルエンジニア(エンベデッドシステム)試験	1
<b>ITベンダーの資格</b>	<b>236</b>
Oracle	99
マイクロソフト	37
Cisco	35
SUN(JAVA以外)	25
SUN(JAVA)	17
IBM(DB2以外)	7
IBM(DB2)	9
SAP	2
HP	2
RHCE(RedHat社)	2
EMC	1
<b>その他団体等の資格</b>	<b>43</b>
LPI(LINUX関係)	18
UMLプロフェッショナル	10
XMLマスター	9
CIW(インターネット技術者の力量を認定する総合教育カリキュラム)	4
PMP資格(プロマネの資格)	2
<b>上記以外ITに関する資格</b>	<b>21</b>
<b>語学関係</b>	<b>11</b>
<b>その他分類(非IT系の資格)</b>	<b>52</b>



## 9. 要望

9 - a - 1. 大学教育への要望

## 9-a-1. 大学教育への要望

回答者: 328名

1. 総論的	人数
1) 産業共同研究の充実	7
2) 産業共同での教育	10
3) 産学の連携	2
4) 技術者育成	6
5) 研究・調査をより高く	4
6) 産学の壁の撤廃	1
7) 産・学の中間の教育の場を	1
8) 教育改革を	4
9) 人格教育	2

2. 教育内容、方法	人数
1) 海外経験	6
2) 講義と演習のバランス	22
3) 実践から理論・基礎を学ぶ	63
4) インターンシップ(国内外)	14
5) 最新技術に則した教育	21
6) 学問、技術の必要性教育	15
7) 技術者の講師	8
8) 英語力(専門書、会話)	30
9) 文章力、プレゼン力	11
10) コミュニケーション力	10
11) 幅広い知識	10
12) 理論・基礎の充実	23
13) 科目の構造化での教育	9
14) 専門、理解力を深める	5
15) ソフトウェア工学	16
16) グループ開発	8
17) ハード、ソフト製作	1

3. その他	人数
1) プログラミング、スキルアップ	16
2) 教育への熱意、方法アップ	2
3) 自主的学習能力アップ	8
4) 論理的思考能力アップ	3
5) 資格取得コースの設定	2
6) 現状でよい	2
7) 卒業を難しく	3
8) その他	各1件
・ 大学情報研究者への不信	
・ 就職の大学支援をすること	
・ 成績だけでなく評価を多面的に	
・ 学科の壁を撤廃せよ	
・ 正しい理論教育	
・ 組み込みソフト	
・ ツール教育	
・ 習熟度別教育	
・ スピード感を養う教育	
・ 独創性を培う教育	
・ 社内教育を参考に	
・ 教育の一貫性	
・ 目的別教育	

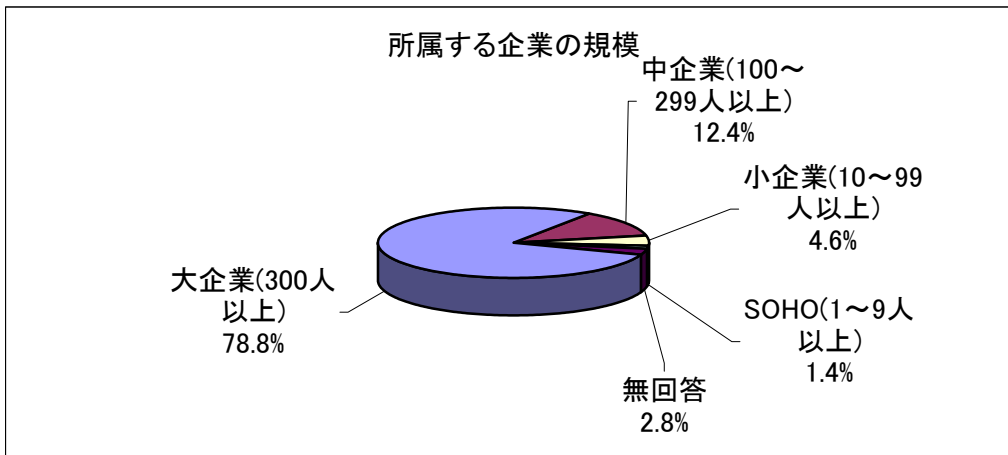
## 10. 所属している企業について

10-a-1. 所属している企業の規模

10-b-1. 所属している企業の業種

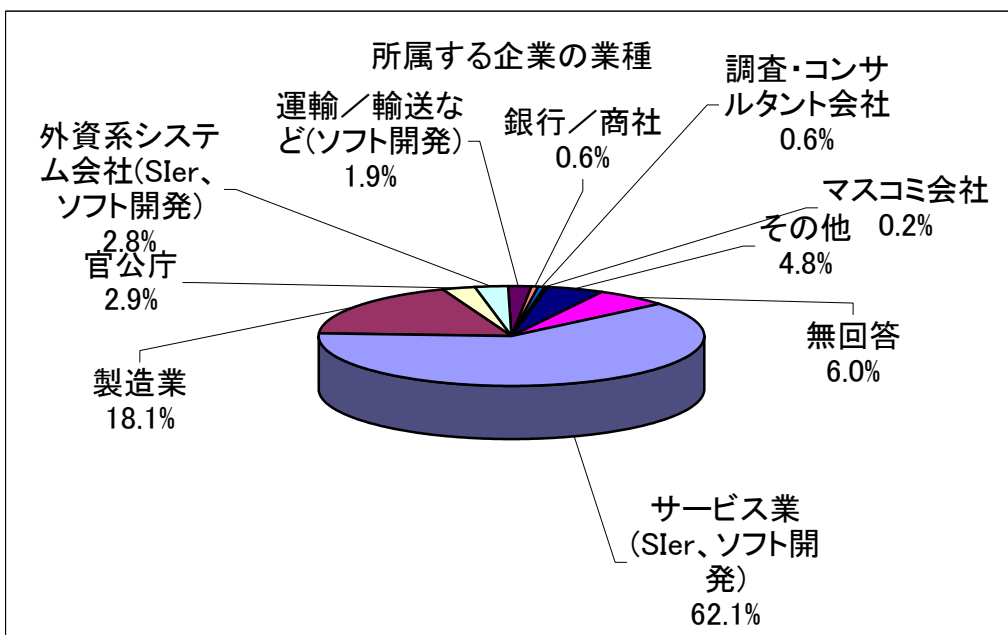
### 10-a-1. 所属する企業の規模

規模	人数
大企業(300人以上)	509
中企業(100～299人以上)	80
小企業(10～99人以上)	30
SOHO(1～9人以上)	9
無回答	18



### 10-b-1. 所属する企業の業種

業種	人数
サービス業(Sier、ソフト開発)	401
製造業	117
官公庁	19
外資系システム会社(Sier、ソフト開発)	18
運輸／輸送など(ソフト開発)	12
銀行／商社	4
調査・コンサルタント会社	4
マスコミ会社	1
その他	31
無回答	39



## 1 1. 所属する学会について

- 1 1 - a - 1. 国内学会への所属数
- 1 1 - a - 2. 国内学会への所属数(詳細)
  
- 1 1 - b - 1. 国外学会への所属数
- 1 1 - b - 2. 国外学会への所属数(詳細)

11-a-1. 国内学会への所属数

所属人数	76
平均数	1.3

11-a-2. 国内学会への所属数(詳細)

所属国内学会数	数
0	570
1	58
2	15
3	2
4	1
5	0

11-b-1. 国外学会への所属数

所属人数	8
平均数	1.3

11-b-2. 国外学会への所属数(詳細)

所属国外学会数	頻度
0	638
1	6
2	2
3	0
4	0
5	0

## 5. 分析と考察

---

本章では、様々な視点からの分析を交え、アンケートの質問項目に沿って順次結果を述べる。適宜、前章に示したアンケート回答の基本データを参照願いたい。なお、基本データを参照する場合、“[]”にて図表番号を表記する。

### 5.1. アンケート回答者

前章の図表[1-a-1~5][1-b-1~3]より分析した回答者の特徴を以下に示す。

1. 大学学部卒、修士卒が各々300名(46.4%)で、両者合わせて600名(92.8%)と大半を占める。博士卒も14名(2.2%)であった。
2. 卒業後の経過年数では、2年以内が254名(39.3%)と多い。就職後に企業活動の厳しさに触れ、学校時代の記憶も鮮明で、その反省を含めて、後述するが、記入式にも積極的な意見が述べられている。
3. 学校時代の専攻は情報系が398名(61.6%)と約3分の2である。下記で触れる現在の業務内容と照らし合わせて、当初の調査目的に沿うものとなっている。
4. 職種は複数回答を含めて、ソフトウェア技術者が408名(63.1%)と多く、システム開発・研究が42名(6.5%)を入れると約70%がソフトウェア/システム開発技術・研究者と考えられる。分析はソフトウェア技術者教育の参考になる。
5. 最終学歴は、国立が271名(42.0%)、私立大学が167名(25.9%)と多い。無回答が175名(27.1%)いるが、ソフトウェア/システム開発に従事している人の大半が大学卒業以上であると推定される。

## 5.2. 講義・演習の方法と効果

ソフトウェア技術者にとって必要である、プログラミング言語、計算機科学、オペレーティングシステム、データベース、ソフトウェア工学等についての理解度、有益性等を調査した。詳細で膨大な貴重なデータを回答者のご協力でご得ることができた。次節以降、勉強方法、理解度、効果（業務に役立っているか）について述べる。

### 5.2.1. 学校別の学習科目の特徴

学校の性格による学習科目の違いを下表に示す。

科目分類	科目	高専卒			学部卒			修士卒			博士卒		
		授業	授業 + 自分	勉強せず	授業	授業 + 自分	勉強せず	授業	授業 + 自分	勉強せず	授業	授業 + 自分	勉強せず
プログラミング入門	プログラミング入門	8	2	1	157	92	25	112	131	18	1	9	
形式言語とオートマトン	形式言語とオートマトン	3		7	120	20	145	172	33	83	7	2	1
プログラミング言語論	プログラミング言語論	4	2	2	132	69	49	116	93	46	4	5	
	プログラミング言語体系	6	1	2	122	44	92	128	49	75	2	5	3
	汎用系	5	2	2	93	32	147	146	50	76	4	2	6
	インタプリタ言語	3	2	4	79	19	151	88	62	89	2	5	3
	C 言語	6	1	3	112	94	54	78	141	22	1	9	
	オブジェクト指向言語	2		5	59	41	136	33	56	88		2	2
	スクリプト系言語	1		6	20	5	222	19	15	136		1	6
	Web 系言語	2		5	17	9	209	8	7	173	1	1	7
計算機科学	計算機科学	7	1	1	128	41	53	149	62	24	3	6	
	基礎理論	8	1		175	49	44	201	58	26	8	5	
	データ構造	7	1	1	152	52	60	166	82	29	5	6	
	アルゴリズム	7	1	1	148	54	62	158	88	27	5	6	
	計算機アーキテクチャ	8	1		150	43	69	155	77	37	6	5	1
情報数理学	情報数理学	4		3	98	33	121	125	46	74	4	5	1
	多次元データ解析	2		6	57	13	196	86	30	153	4	2	3
	パターン認識	2		6	71	17	171	114	32	117	2	4	5
	マルチメディア	3		5	78	31	148	101	47	114	3	2	4
	AI	1		7	53	8	198	106	36	129	2	4	3
	機械学習			8	50	18	192	96	42	129	2	8	1
	最適化手法	3		5	77	17	167	111	44	115	6	4	1
	ファジー			8	53	7	200	87	18	167	4	4	3
	カオス			8	40	5	222	50	16	204	3	2	4
	エージェント			8	10	5	247	38	14	214	1	3	7
	データマイニング			8	22	4	235	37	14	212		1	9
ネットワーク工学	ネットワーク工学	4		3	84	36	114	89	65	59	3	2	4

科目分類	科目	高専卒			学部卒			修士卒			博士卒		
		授 業	授 業 + 自 分	勉 強 せ ず	授 業	授 業 + 自 分	勉 強 せ ず	授 業	授 業 + 自 分	勉 強 せ ず	授 業	授 業 + 自 分	勉 強 せ ず
	プロトコルと伝送制御	5	1	2	103	41	96	101	75	59	4	1	4
	ネットワークアーキテクチャ	6		2	93	42	110	105	62	76	4	1	5
	伝送制御方式	7		1	109	35	102	121	51	77	5	1	5
	符号化と伝送	7		1	115	33	98	139	59	62	6	1	4
	LANとWAN	7		1	97	43	101	101	57	72	4	1	5
	インターネット応用	3	1	3	59	27	153	54	48	123	2	1	6
オペレーティングシステム	オペレーティングシステム	6	1	1	116	48	80	108	89	37	5	6	1
	コンパイラの役割と働き	5	1	3	136	36	86	154	58	49	5	5	3
	UNIX	4		4	110	44	100	94	99	50	3	7	1
	LINUX	3		4	28	16	197	9	28	149			5
	WINDOWS	4		1	54	44	115	12	36	105			3
	MAC OS			8	11	7	235	4	6	224			8
	TRON			8	2	4	261	3	3	270			12
	Main Frame OS	1		6	5	4	260	8	3	271			12
	その他			1		1	65	1	1	50			1
データベース	データベース	5	1	2	85	29	134	93	45	109	3	1	8
	データベースの原理・理論	5	1	1	94	25	126	114	35	101	3	1	9
	データの分析と設計方法	4		2	66	26	153	87	30	131	3	1	8
	データ操作	4		2	73	23	138	72	34	118	3		8
	DBMSの機能と特徴	4	1	2	51	23	167	80	25	135	2	1	8
	分散データベース	2		4	31	14	202	53	14	190	1		10
	データベース設計論	3	1	3	34	8	200	65	17	173	2	1	9
	汎用機のDB	2		5	10	7	237	8	5	253			12
	リレーショナルDB	2		5	33	22	184	42	23	163	1		9
	オブジェクト指向DB			8	15	7	232	10	4	247			12
	その他			1			75		1	64			2
ソフトウェア工学	ソフトウェア工学	2	2	2	81	30	120	97	41	92	3		7
	ソフトウェア記述言語	5	2	1	95	34	119	116	47	87	3		5
	ソフトウェア開発プロセス	1	2	4	58	26	163	74	32	144	1		9
	プロジェクトマネジメント	2		5	27	8	214	26	7	232		1	11
	ソフトウェア要求定義	2		5	35	10	201	42	18	196		1	10
	ソフトウェア設計方法論	3		4	40	14	194	50	20	185	1	1	8
	ソフトウェア構築方法論	3		4	35	10	201	39	19	200	1	1	8
	ソフトウェア品質	2		5	38	10	198	43	15	197	1		10
	ソフトウェアテスト	2		5	30	10	203	37	19	199			11
	ソフトウェア保守	2		5	29	10	206	22	12	227			11
	ソフトウェア構成管理	2		5	28	8	210	18	11	235			9
	ツール及び手法	3		4	23	12	207	19	18	218			10

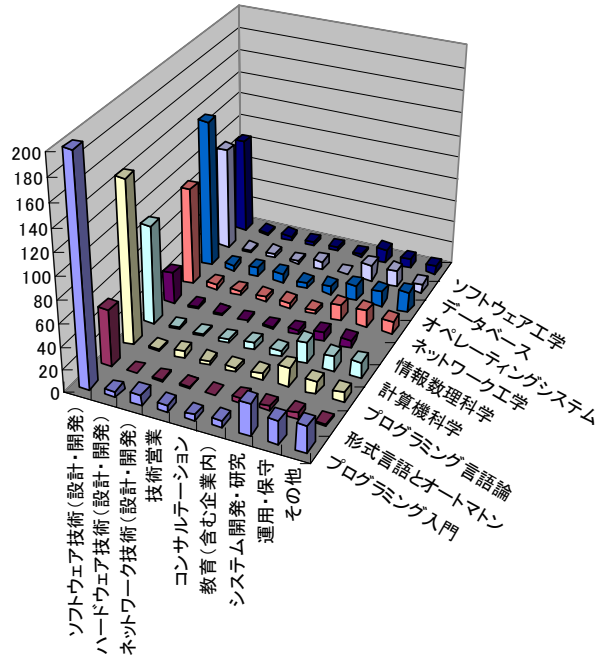
以下に特徴を述べる。

1. 高専での履修科目は「プログラミング入門」、「計算機科学」関連が多く、「情報数理科学」関連、「プログラミング言語論」関連が少ない。未履修科目は「情報数理科学」関連、「プログラミング言語論」関連、「ソフトウェア工学」関連が多い。
2. 大学学部、及び大学院での履修科目は「プログラミング入門」、「計算機科学」の「基礎理論」「データ構造」「アルゴリズム」、「プログラミング言語論」の「C言語」が多く、「オペレーティングシステム」関連、「プログラミング言語論」の「スクリプト系言語」「Web系言語」、「ソフトウェア工学」関連が少ない。この場合、情報関連学科以外の卒業者が約40%いるので、より詳細な分析が必要である。しかし、未履修科目のソフトウェア工学については、今後大学の課題である。
3. 博士卒についても、学部・大学院とほぼ同様で、履修科目では特に「計算機科学」の「基礎理論」「計算機アーキテクチャ」が多い。
4. 勉強方法で見えていくと、「プログラミング入門」「計算機科学」「形式言語とオートマトン」「ネットワーク工学」は80%以上で授業がある。一方、「プログラミング言語論」「情報数理科学」「オペレーティングシステム」「データベース」「ソフトウェア工学」は授業で取り扱っていないということがわかる。「プログラミング入門」「プログラミング言語論」は自分で勉強したことがある人が多く、独学で学びやすいものだと思う。
5. 理解度で見えていくと、「情報数理科学」「オペレーティングシステム」「データベース」「ソフトウェア工学」は「理解できなかった」という回答が、約40～50%もある。いずれの科目も「十分理解した」というレベルに達している人は少ない。
6. 効果の面では、「形式言語とオートマトン」「オペレーティングシステム」「データベース」「ソフトウェア工学」は「役立っていない」との回答がいずれも約50%あり、「情報数理科学」においては約70%に至っている。「十分役立っている」という人は少なく、学校で教えていることと、企業で必要とされる知識との間のギャップが気になるところである。

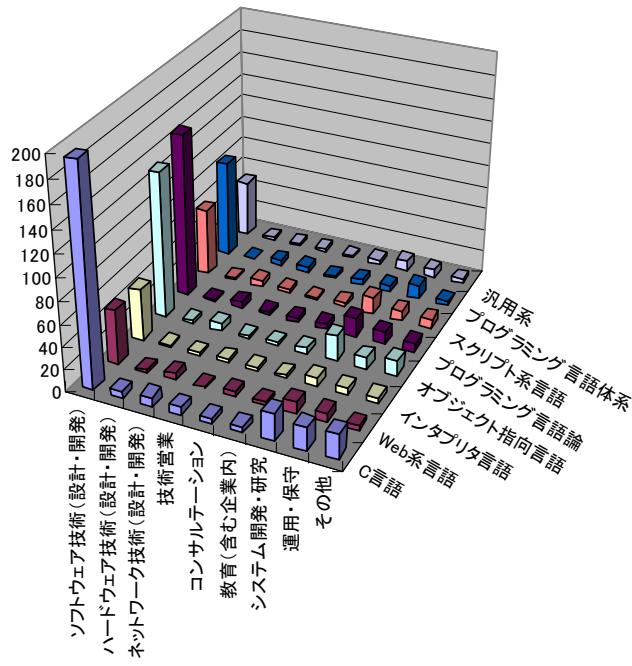
## 5.2.2. 職業別の役立っている科目

現在の職種別に、学校時代の履修科目が役立っているかを、現在の業務についての調査結果と組み合わせ分析している。下図に結果を示す。

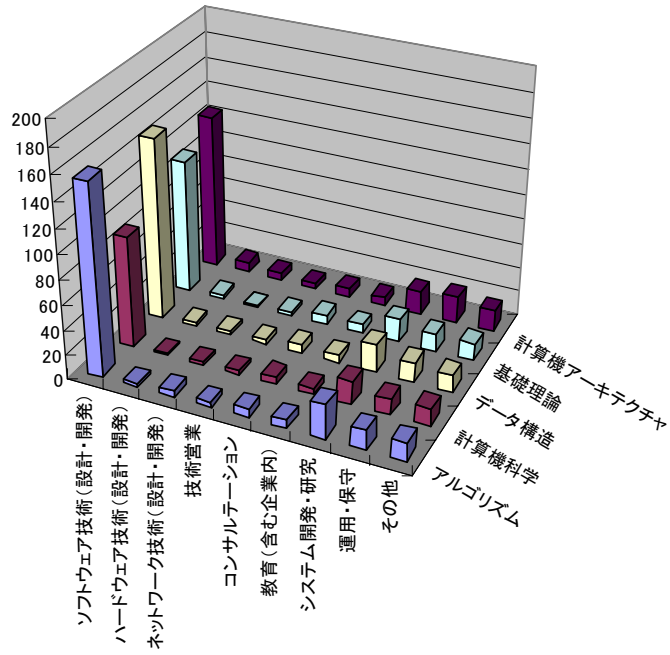
大分類科目での役立ち



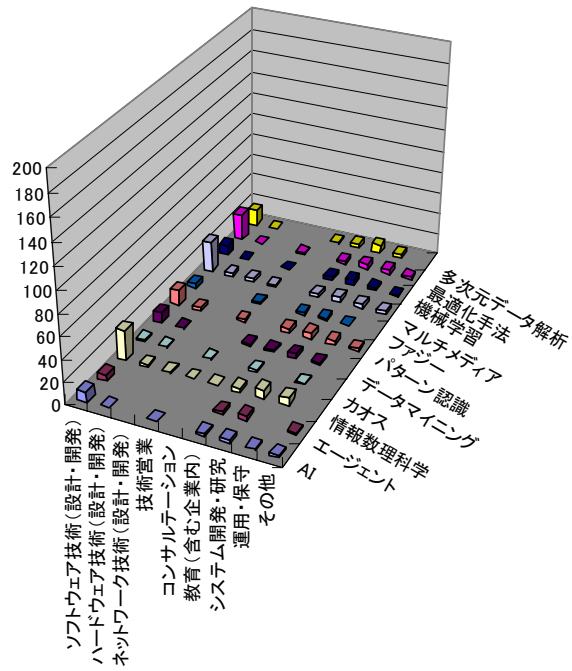
プログラミング言語論の科目毎の役立ち



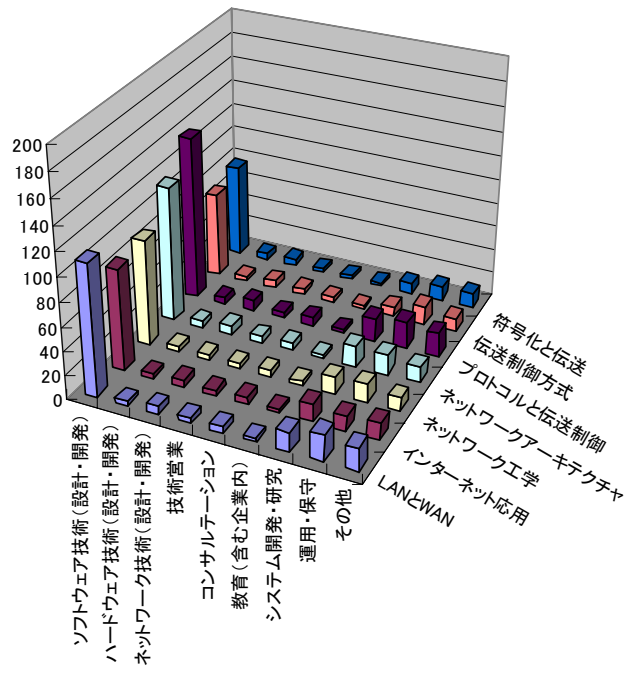
計算機科学の科目毎の役立ち



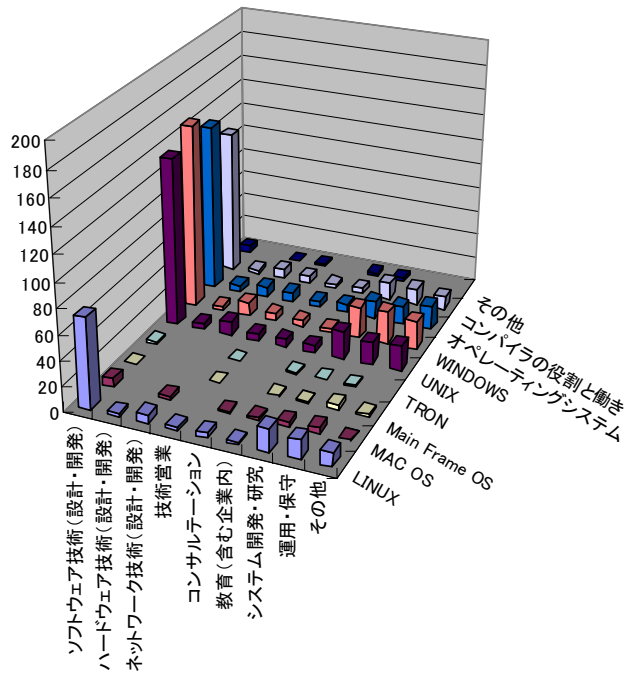
情報数理工学の科目毎の役立ち



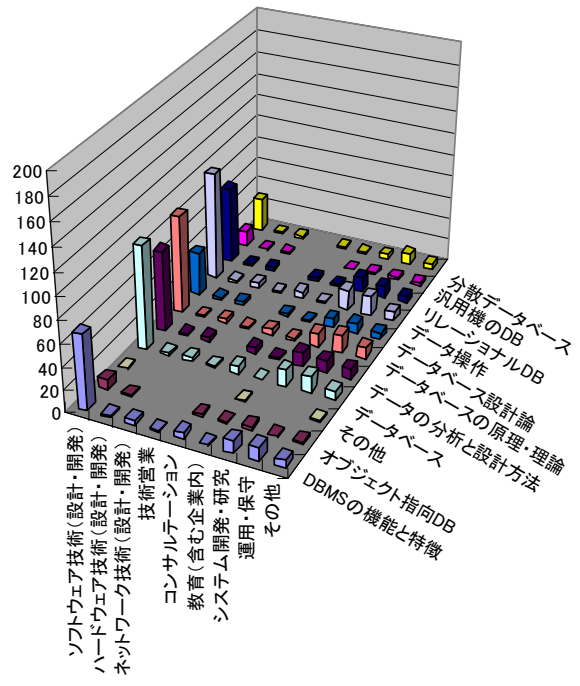
ネットワーク工学の科目毎の役立ち



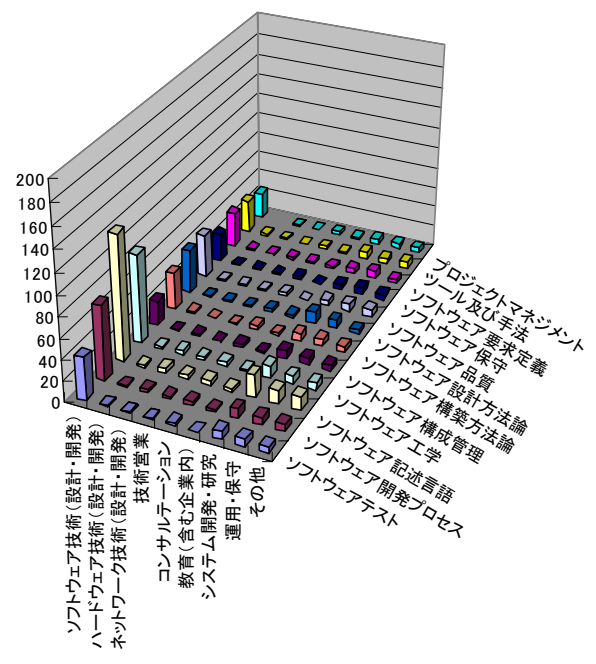
オペレーティングシステムの科目毎の役立ち



データベースの科目毎の役立ち



ソフトウェア工学の学科毎の役立ち



上図の特徴的な部分を以下の二つの表に示す。ソフトウェア技術者にはプログラム言語などの授業が役立っているが情報数理学は役立っていないことが分かる。

◆ 「ソフトウェア技術者」で「役立っている」「十分役立っている」が多い科目

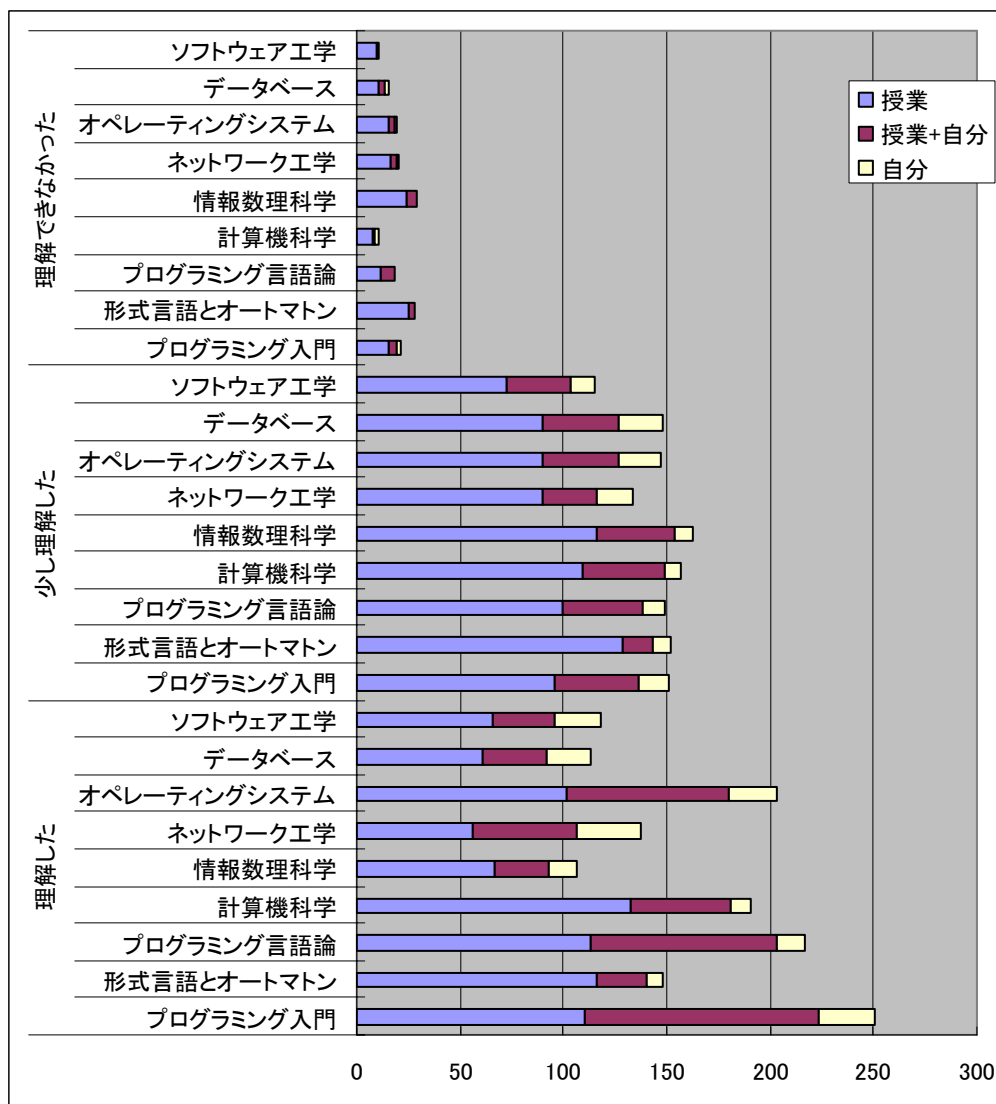
科目	科目分類	十分役立っている	役立っている	合計
プログラミング入門	プログラミング入門	99	104	203
C 言語	プログラミング言語論	117	77	194
アルゴリズム	計算機科学	64	90	154
データ構造	計算機科学	59	88	147
WINDOWS	オペレーティングシステム	84	63	147
プログラミング言語論	プログラミング言語論	45	101	146
プロトコルと伝送制御	ネットワーク工学	52	84	136
UNIX	オペレーティングシステム	78	57	135
オペレーティングシステム	オペレーティングシステム	51	83	134
オブジェクト指向言語	プログラミング言語論	83	48	131
計算機アーキテクチャ	計算機科学	44	82	126
ソフトウェア記述言語	ソフトウェア工学	55	66	121
コンパイラの役割と働き	オペレーティングシステム	45	70	115
ネットワークアーキテクチャ	ネットワーク工学	34	79	113
基礎理論	計算機科学	31	78	109
LANとWAN	ネットワーク工学	33	76	109
データ操作	データベース	49	48	97

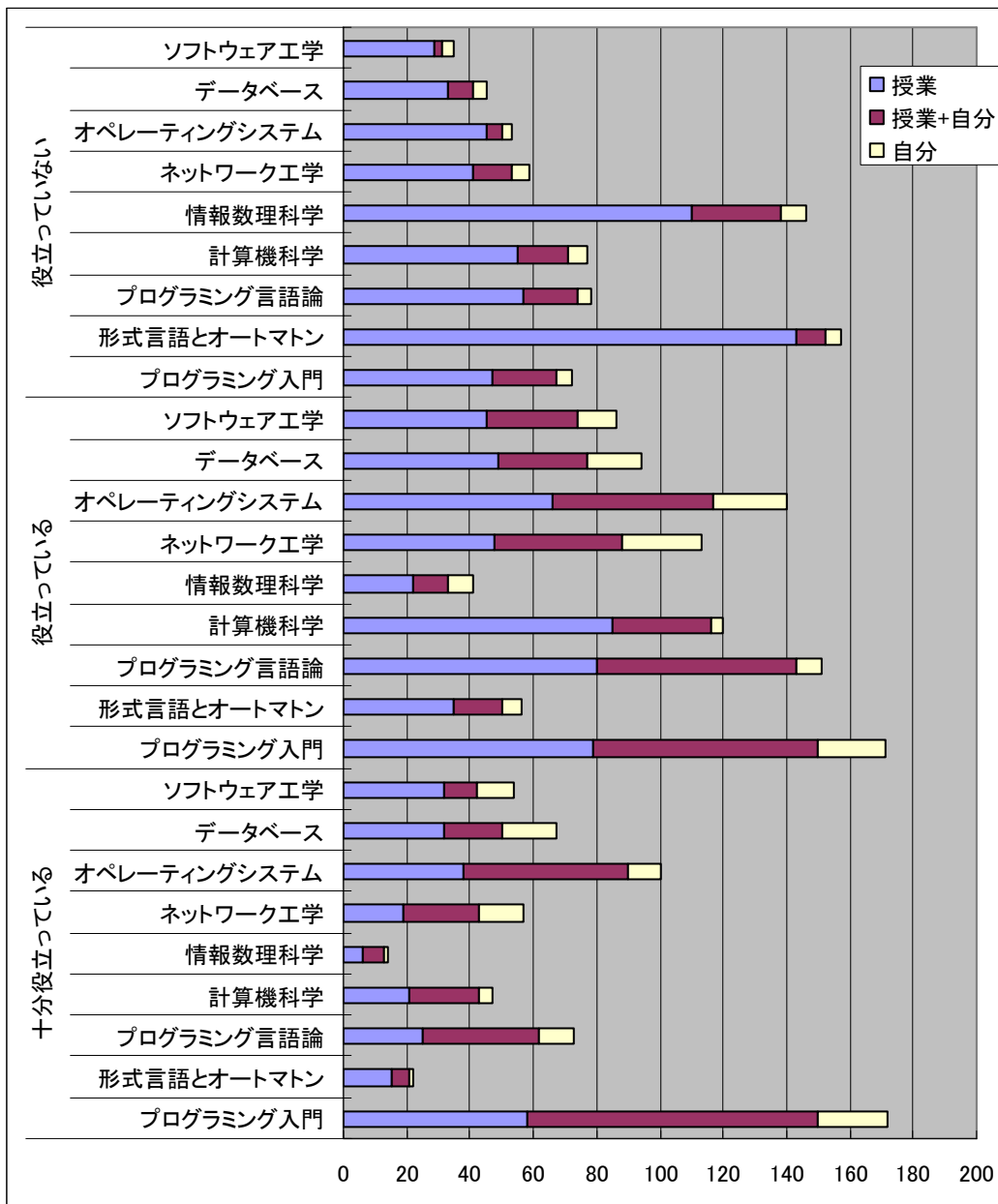
◆ 「ソフトウェア技術者」で「役立っていない」が多い科目

科目	科目分類	役立っていない
機械学習	情報数理学	166
AI	情報数理学	164
カオス	情報数理学	162
ファジー	情報数理学	160
パターン認識	情報数理学	158
多次元データ解析	情報数理学	155
インタプリタ言語	プログラミング言語論	154
MAC OS	オペレーティングシステム	151
最適化手法	情報数理学	149
TRON	オペレーティングシステム	148
エージェント	情報数理学	142
汎用系	プログラミング言語論	141
マルチメディア	情報数理学	141
Main Frame OS	オペレーティングシステム	137
情報数理学	情報数理学	131
形式言語とオートマトン	形式言語とオートマトン	131
データマイニング	情報数理学	130

### 5.2.3. 科目別特徴について

アンケートでの大分類 9 科目について分析するために下図を作成した。



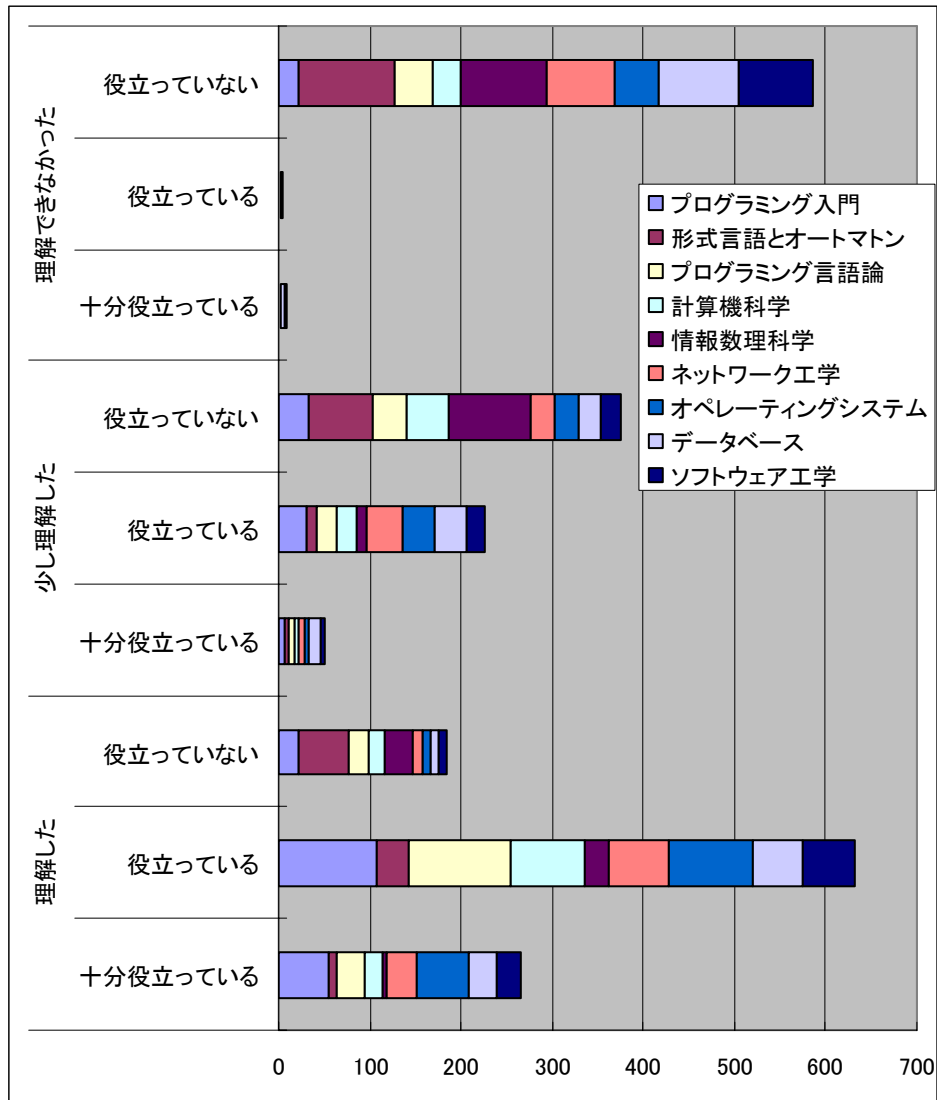


上図より分析できた特徴を以下にまとめる。

1. 科目別で勉強法と理解度の関係では、授業と自学習を行っている人で理解度が増している。有益性でも同様である。各科目の特徴は、「プログラミング入門」「プログラミング言語論」「オペレーティングシステム」「ネットワーク工学」で授業と自学習での理解が高い。有益性では、これらに加えて「オペレーティングシステム」で有益性が高くなっている。

## 5.2.4. 理解と役立ち

授業や自学習での理解の度合いが、現在の業務における効果とどのような関係にあるかを下図に示す。



上図より、理解したものについては役に立っているが、少しの理解では必ずしも十分に役に立っていないようである。現業で役立つためにはしっかりとした理解が必要と言える。

## 5.2.5. 勉強方法と理解

以下、勉強方法と理解について大分類の科目別に特徴を述べる。

### 1. 勉強方法と理解：プログラミング入門

授業のみでなく、自分でも勉強することによって、「少し理解した」という曖昧なラインの割合と「理解できなかった」ということが減る。

### 2. 勉強方法と理解：形式言語とオートマトン

「授業のみで勉強した」との回答が多く、自学習はかなり少ないようである。一方、「少し理解をした」「理解できなかった」が約50%を占めており、授業の比率に比して理解度の向上に課題があるかと思える。

### 3. 勉強方法と理解：プログラミング言語論

プログラミング言語の基礎については多くが学校で授業があり、60%近くが理解している。言語は「汎用系」「C言語」について授業があるが、「オブジェクト指向言語」「スクリプト言語」といったものに関しての授業は少なく、独学のみという人が多い。授業には無いのに、それに関心を持って独学で勉強している人が多いという現状がうかがえる。

### 4. 勉強方法と理解：計算機科学

ほとんどが授業で取り扱われている。独学のみという人は少ないようである。

### 5. 勉強方法と理解：情報数理学

ほとんどの人が十分な理解に至っていない。理解が難しい科目であるということだろうか。「勉強せず」との回答も多い。

### 6. 勉強方法と理解：ネットワーク工学

自学習を含む場合は、理解の比率は高いが、授業のみの場合は「少し理解した」との回答が約半数に上る。指向性の高い分野のようである。

### 7. 勉強方法と理解：オペレーティングシステム

具体的なOSについて学校側では、UNIXは取り扱っているが、LinuxやWindowsなどのその

他の OS については取り扱っていない。この回答は「自分で勉強した=使ったことがある」「勉強せず=使ったことがない」というような結果とも考えられる。

8. 勉強方法と理解：データベース

授業が「ある」と「ない」で半々に分かれる結果となった。データベースについて取り扱っていない学校が（他の科目と比べて）多いようである。

9. 勉強方法と理解：ソフトウェア工学

授業で取り扱っている学校が少ない。自分で勉強している人の割合が他の科目と比べて多いことから、学校ではやらないが重要な科目ということなのではないだろうか。

## 5.2.6. 勉強方法と効果

以下、勉強方法と効果について大分類の科目別に特徴を述べる。

1. 勉強方法と効果：プログラミング言語

約 53% が役立っている。「少し役立っている」を含めると約 79% になる。授業で取り扱われている比率も非常に高い。

2. 勉強方法と効果：形式言語とオートマトン

多くの学校で授業として取り扱っているようであるが、「役立っている」の回答は極めて少ない。不要であるとは言わないが、必修である場合が多いと思われるので、選択科目にするなどの検討を要するのではないだろうか。

3. 勉強方法と効果：プログラミング言語論

授業で取り扱われているものでも、「プログラミング言語論」「C 言語」は役立つが、その他は役に立っていないという回答が多い。「オブジェクト指向言語」「スクリプト言語」「Web 系言語」は独学で学んでいる人が多く、これらは実際に役立っているとの回答も多い。しかし、授業で取り扱われているのは少ないようである。

4. 勉強方法と効果：計算機科学

授業で学んだ人は多くるが、独学は少ない。「役立っている」との回答は約 50% ある。「役立っていない」ということは約 10% 程度で、極端に少ないわけではない。

5. 勉強方法と効果：情報数理学  
授業で多く取り扱われてはいるものの、「役立っていない」との回答が約50%以上を占めている。「役立っている」「十分役立っている」との回答は、約10%程度である。
6. 勉強方法と効果：ネットワーク工学  
「役立っている」との回答は約50%程度ある。授業でも比較的多く取り扱われているようである。
7. 勉強方法と効果：オペレーティングシステム  
授業で取り扱われている科目では約50%の人が役に立っていると答えている。授業での取り扱いが比較的少ない、Linux や Windows に関しては約80%が「役立っている」との回答である。自分で色々使ってみて、使い方を覚えるのが一番だが、学校側のサポートが必要でないだろうか。
8. 勉強方法と効果：データベース  
授業で取り扱われているものは50%程の人が「役立っている」と回答している。勉強していない人が多いが、独学で勉強している人の多くが「役立っている」と答えているのが目立つ科目である。勉強していない人＝授業でも習っていない、なので、もっと授業に組み込むべき科目の一つではないだろうか。
9. 勉強方法と効果：ソフトウェア工学  
授業で取り扱われているものは50%程度の人が「役立っている」と答えている。授業で取り扱われていない科目が多い。これらについては、授業と自学習を問わず勉強していれば「役立っている」との回答が50%程ある。授業で取り扱われていない科目の中に、授業に組み込むべき科目があるようである。

## 5.2.7. 印象に残っている科目

印象に残っている科目については、「情報工学」という回答が約30%を占める。続いて約7%が「専門科目」と答えている。意外にも専門科目に対する印象が薄いようである。

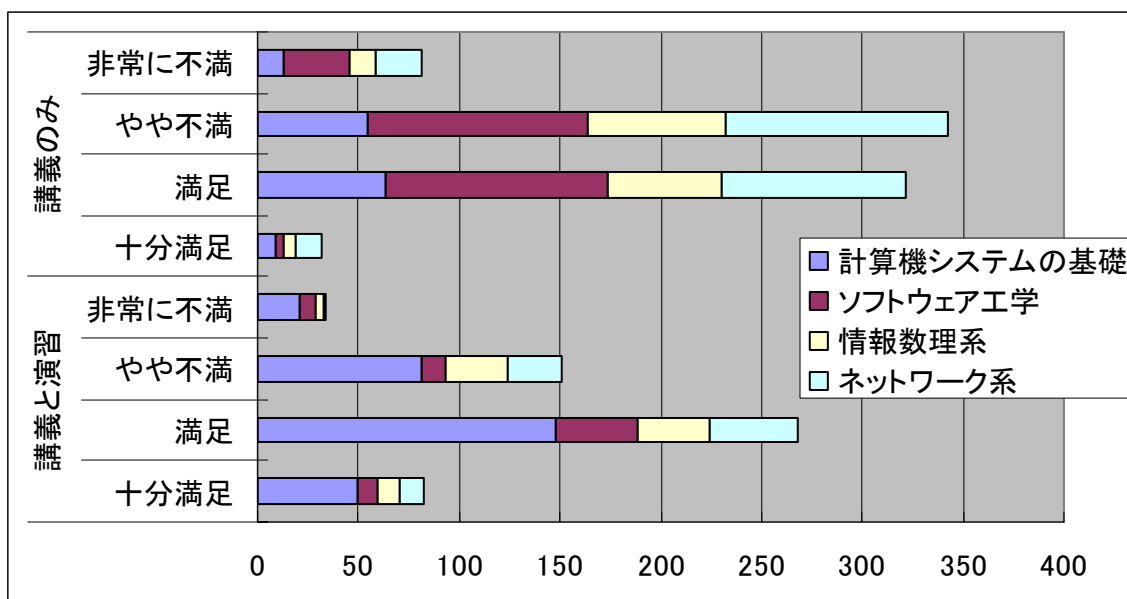
印象に残っている理由の三大回答は「理解しやすい」「新しい知識の獲得」「興味があった」である。しかし、これらを合わせても全体の約31.6%であり、まだまだ理解の浸透率は低いようである。入学者は当然、情報工学に興味があり入学してくるわけであるから、その興味ある情報工学の授業において、新しい知識をもっと理解しやすく教える必要があるようだ。

また、印象に残った理由で、「面白い、興味がある」「新しい知識・概念・方法を学んだ」で約22.2%、「教官が良い」は約3.4%であった。これは、将来の職務に近いことに興味を引かせ、分かり易く、新しい概念・知識・方法論を講義・演習で行うといった、教育方法としてオーソドックスとされていることが一番大切であることを想起させる。

## 5.3. 講義・演習の方法と満足度

### 5.3.1. 授業方法と満足度

授業方法と満足度の関係を下図に示す。「講義と演習」の場合は、「十分満足」「満足」を含めて高い満足度を示している。一方、「講義のみ」の場合は、「やや満足」が多く、「十分満足」が少なくなっている。特に、「計算機システムの基礎」で、「講義と演習」での満足度が高く、「講義のみ」の「ソフトウェア工学」「ネットワーク工学」で満足度が低いのは特徴的である。



以下、[3-1-1-a-1~20][3-1-1-b-1~20]について分類毎に述べる。

#### 1. 計算機システムの基礎：

講義や演習は他に比べて多く用意され、演習の割合は半分程度も行っている。全体的に満足度が高く、「不満」は非常に少ない。特に「プログラミング基礎」が最も多く演習と講義を行っており

約69%、その満足度は「十分満足した」「満足した」を合わせると55%と高い。一方、「計算機概論」は「講義のみ」が多く、「非常に不満」が他より若干多い傾向がある。

2. ソフトウェア工学関連：

ソフトウェア工学関連は授業が少なく、またあるにしても講義中心であるものが多い。特徴として「不満」の回答率が高い。特に「プロジェクトマネジメント」では、ほとんどの人が講義も演習もなく（71%）、満足度も回答者の中で「非常に不満」と答えた人が60%以上もいる。

3. 情報数理系：

情報数理系もまた、授業が少なく不満に思っている人の割合が高い。

4. ネットワーク系：

ネットワーク系の勉強方法は、「通信ネットワーク基礎」「通信方式」「信号処理」などの質問項目に対し、どれも同じ比率を示している。また、講義と演習が13%程度と他と同等程度の割合となっている。

全体的に演習を含んだ講義が少ないことが挙げられる。科目によってはほとんど授業がないこともある。満足度との関係で見ると演習の割合が多いほど満足度が高いという傾向が見られる。

## 5.3.2. 授業科目と過不足

以下、[3-1-2-a-1~4]について順次述べる。

1. 計算機システムの基礎：

演習を行う授業が必要だと回答している人が最も多いといった特徴がある。前項の分析で他の科目に比べ多くの方が演習を行ったと回答しているが、本回答を鑑みるに、より多くの演習が必要だと言える。「コンピュータアーキテクチャ」も比較的要望が高いようである。

2. ソフトウェア工学関連：

追加科目の意見として229名もの回答があり、ソフトウェア工学の授業に対する期待がうかがえる。前項の分析で授業を受けていないと回答した人が多かったプロジェクトマネジメントでは44名が「追加して欲しい科目」で多く挙がっている。このプロジェクトマネジメントは「追加して欲しい科目」の中でも具体性が高い記述が多く、学生の段階から必要な知識として強く認識

されているようである。また、システム開発演習も63名と要望が多く、実際の作業でしか学べないことも多いと認識されているので、今後実践的な学習方法の検討が必要であろう。

### 3. 情報数理系：

情報の基礎的な内容が重要であるとの回答と、応用的な内容が重要であるとの回答に分かれている。このことは基礎知識をしっかりと固め、応用可能なレベルの理解が必要と認識されていると考えられる。そのため、理解がしっかりしたものであるかを確かめる方法を取り入れ、段階的に応用へ進んでいく授業体系が望まれる。

### 4. ネットワーク系：

ネットワーク系ではネットワークの構築演習を求める意見が72名と多く、実際に体験し理解を深める授業が望まれている。プロトコルをはじめとするネットワークの基礎からしっかりと理解して欲しいとの要望も寄せられ、今後、ネットワークの構築を基礎から行う授業を多く取り入れていくべきであろう。

全般的に「追加して欲しい科目」は多く回答されているが「不要な科目」は少ない。これは学校教育において情報関連の授業が全体的に不足しているのではないかと感じているのだろう。特に演習を用いた授業が望まれ、より実践的な教育が求められている。回答者は実践的な教育の効果を強く感じているのだろう。

## 5.3.3. PBL

学校教育においてグループで実演習を行うPBL(Project Based Learning)が多くなっている。しかし、教官への負荷が多いことや基礎・原理をしっかりと勉強しておかなければならないことも強調しておく必要はある。その効果を確認するためにこのアンケートを実施した。結果はPBLの経験があるとの回答は87名で全体の約13%であった。

テーマとして最も多かったのが設計から開発までを行う企業の業務的な内容である。例えば、Javaを用いたソフトウェア開発やデータベースの設計開発などがある。次に多いのが実験を行う研究的な内容である。これにはシステムの分析演習などもあるが、カオスを用いた研究なども見られる。

PBLへのご意見は、多くが好意的である。良いという意見を集約すると、グループ単位で作業を行うためコミュニケーションの必要性や個人の役割の把握など、様々な課題と直面することで、自ら考えることやグループ作業の難しさなど多くのことが学べることを期待している。

一方、良くないという意見も若干ある。それは、学校で行う PBL ではグループ作業とはならず特定の人だけの作業になる危険性があるという意見である。

以上の意見を考慮し、学校側が行う授業として、学生全員が参加できるような仕組みでグループ学習を行える実践型学習を進めていくべきだと考える。

### 5.3.4. インターンシップ

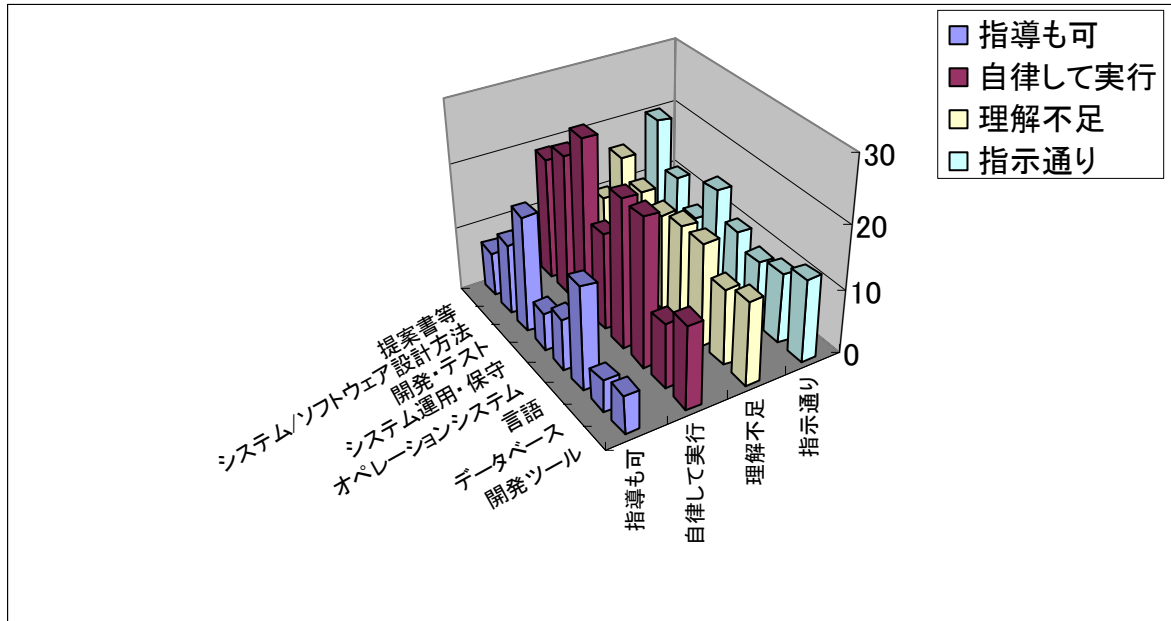
インターンシップ制度の導入については文部科学省が推進していることもあり、教育現場にも広まってきている。しかし、1週間から1ヶ月程度の短期間であり、企業活動の表層を見るだけである。しかし、それを学習効果のある教育とするためには実業務の経験がより有効と思われる。欧米では自己の専門に近い分野のアルバイトやインターンシップを学部のはじめより行い、個人のキャリアとして評価されている。学問と実学を繰り返し経験し、個人の能力向上に有益になる方法がインターンシップの目的のひとつである。以上の考え方によりアンケートをとった。

インターンシップの経験があると答えた人は22名で全体の約3%と少ない。そのうち17名が1ヶ月のインターンシップ期間と回答している。インターンシップで行う内容はソフトウェア開発がほとんどでJavaを用いたものが多い。他にはソフトウェア開発工程を学んだり研究を共に行ったりというものもあった。

インターンシップに対する意見としては、「実際の現場を見ることで企業の雰囲気を感じられる」「自らの目標とすることが明確になる」というものが多い。また、「研究の刺激になる」との意見もあるなど好意的なものが多い。一方、「現場を見ることで大学教育の無意味さを感じる」という意見や「インターンシップは擬似的なものに過ぎず意味がない」との意見もある。

### 5.3.5. PBL、インターンシップでの習熟

PBL、インターンシップを通しての科目毎の習熟度を下図に示す。



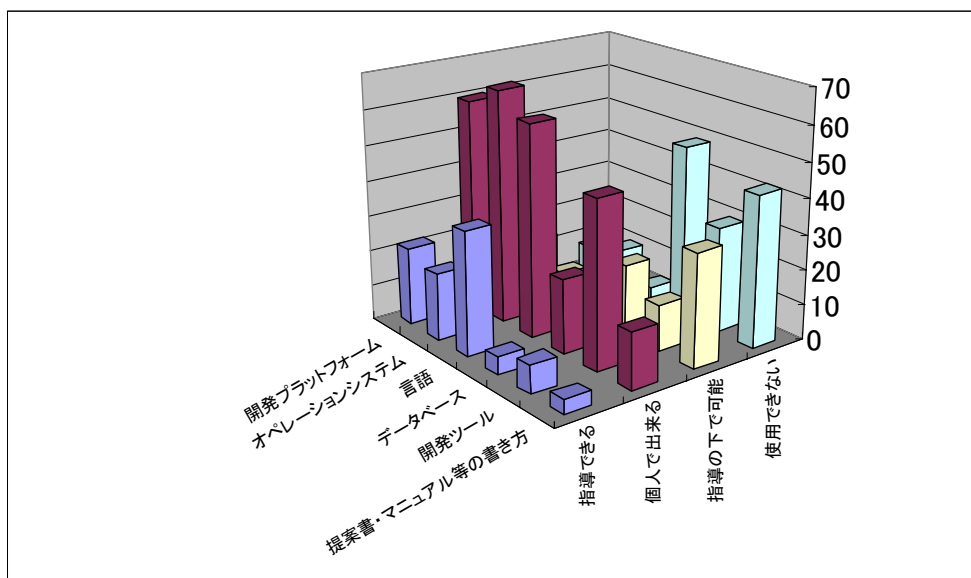
この図から「自律して実行」が比較的高い傾向が出ているが、「理解不足」と「指示通り」の回答を合わせると非常に高い数値になる。PBL、インターンシップの目的を技術習得におくのであれば、その方法については検討が必要かと考えられる。

「開発・テスト」「言語」といった科目では「指導も可」といった高い習熟度を示している。一方、「システム運用・保守」「データベース」といった基盤系技術での習熟度が比較的低くなっている。専門性が高く、環境的にも習得の困難な課題に対して、どのように教育を行っていくかが課題と考えられる。

### 5.3.6. 授業以外でのシステム開発経験

授業以外でのシステム開発経験は、アルバイトが最多で81名（約12.5%）である。全体では、経験者が156名で、約24.1%の人が経験している。

下図は、習得度について科目毎にまとめたものである。



PBL、インターンシップと比べ、「指導の下で可能」「使用できない」は低くなっている。この活動自体が自主性に基づくものであり、習熟度が高いのも理解できる。

一方、科目別に見たときには習熟度に偏りが見られる。「データベース」「提案書・マニュアルの書き方」で非常に習熟度が低下している。これらの技術は、「言語」「オペレーティングシステム」「開発プラットフォーム」に対し、習得のしづらい環境であることがうかがえる。

### 5.3.7. 卒論・修論・博士論文

卒論・修論・課程博士論文に関して、①課題分野（理論的、実システムの、その他）、②研究課題テーマ名、③課題内容、④現時点での感想に関して自由記述形式でアンケートを行った。なお、本項目に関する実データはここに記載しない。その理由は、②の研究課題テーマ名から、論文を作成した場所や指導教官の推論が可能になる危惧からである。ゆえにここでは卒論・修論・博士論文の類別化をせずに、これらを一体化した総論としての結果を報告する。なおここであげる数値の正当性を確保するための元データは、しかるべき機密保護の下に保持される。また、本アンケートの調査の主旨からすると②の内容を明示することよりも、卒論・修論・博士論文作成経験を現時点から振り返って関当事者たちがどのように認識、意識しているかのデータがより重要であろう。以下、順次結果について述べる。

回答者総数：546名（回答率：約84.5%）

①課題分野に関して

- ・ 理論的課題：150件（約23.2%）
- ・ 実システムの課題：207件（約32.0%）
- ・ その他：189件（約29.3%）

これは②とも関係するゆえ実際には②を明らかにしたうえで分析したほうが好ましいかもしれない。特に、その他が約29.3%と多数を占めるのは、IT関連分野が理論的課題と実システムの課題の中間部分が多いことを意味している。記述データから分かる特徴は、OS関係が意外に少ないこと、言語関係はどちらかという使い方の特化していること、アーキテクチャに関する研究課題が皆無状態であること、オブジェクト指向関係も少ないこと、モバイル関連の理論的、実システムの課題も少ないこと、などが目立つ。

④に関して；課題の内容（理論・技術）が実務に役立ったか

- ・ 実務に役立つ：86件（約13.3%）
- ・ 役立たない：380（約58.8%）
- ・ 少しは役立つ：80（約12.4%）

卒論・修論・博士論文の内容が直接的に実務に役立っていないとのデータが約71.2%を占めることは大きな意味を持つ。なぜならばこのデータは指導する側からのデータではなく当事者たちからのデータだからである。ただしこの「実務に役立つ」のカウントには、基礎的勉強をした、あるいは基礎的知識が習得できた、などをも含んでいて、厳密に実務に役立っているとのデータそのものは10%を下回る。このデータの持つ意味をどのように解釈すればよいのだろうか。一概に学側における課題そのものが、企業から見てナンセンスであると決め付けるわけにはいかない難しさがある。特に修論、博士論文では、その内容が先端的であればあるほど、実務分野でこれらを活用しきれていないケースも想像される。このことは留意されるべきであろう。しかし現状における当事者たちからの実データの持つ意味は大きい。

④のなかで、実務以外に役立つとの以下の答えの中身を数値化するのは困難であり、数値化そのものが意味を持つとも思えない。それらは、(a)やったという精神的満足感、(b)実験計画・企画、(c)プレゼンテーション、(d)纏め方、(e)ものの考え方、に纏められる。この傾向は次節の「学会・論文の経験と効果」でも同様である。

### 5.3.8. 学会・論文の経験と効果

まず、学会への参加の様相についての結果を[3-5-a-1]に示す。

国内外の学会への参加回数はグラフに示す通りである。回答者の約43%の275名が国内学に参加し、多くの卒業生が1、2回の参加経験を持つことが伺える。中には博士課程の院生が10回以上の参加経験を持つ場合もある。これを発表回数と比較すると205名の卒業生が経験している。10回以上のケースを除いて参加回数と発表回数は相似形を構成している。国外の学会への参加者は約8%にもおよぶ。さらに6%の卒業生が発表経験を持つ。これらの数値の持つ意味をどのように解釈できるであろうか。学会活動の中心がこれら卒業生に依存していることも伺える。学会への参加を促すためには学会への参加費用の低減化も必要であろう。

次に、学会活動の及ぼす教育効果についてのデータを纏めた結果を[3-5-a-2,3]に示す。なお、効果についての分析は回答のあった数を母数としている。

本質問に関する約47%の回答があった。学会活動の持つ教育効果は非常に効果的である(約23%)を含め約83%が効果ありと評価している。一方、約17%が否定的である。高評価の約90%の中身を概括すると知識獲得、プレゼン能力向上などが効果的と判断する根拠である。生データを表[3-5-a-4]に記載する。

国内外の学術誌への論文掲載経験とその効果を調査してみた結果を[3-5-b-1]に示す。

まず、全回当者の22%(141名)が国内外の学術誌への研究論文掲載経験を持ち5回以上の業績を有する卒業生もいる。国内では135名、全体の21%の論文掲載経験者があり、30名は国外学術誌への論文掲載経験を持つ。国内学会での発表経験者との比較でもわかるように発表件数の約半数の研究は論文として掲載されていることがわかる。これと比較して国外での発表件数と論文掲載数はほぼ同数であり、発表論文が掲載論文として採用されているのが伺える。

次に論文掲載についての教育効果についての調査結果を纏めたものが表[3-5-b-2,3]である。なお、効果についての分析は回答のあった数を母数としている。

研究発表の場合とは異なり、肯定的な評価は23%に過ぎない。約77%がその教育的効果を否定していることは注目に値する。唯一の教育的効果として評価されているのは文章力の向上のみである。これは企業において研究論文掲載実勢が実務における業績としてカウントされていないことを示唆している可能性がある。ここにも産学間の「研究業績」そのものに関する認識の乖離現象が伺える。生データを表[3-5-b-4]に記載する。

調査から明らかになったことは学会活動（発表、論文掲載）に関する期待の仕方の産学間の埋めがたい距離である。ここでの主題からは外れることになるが、調査結果は間接的には学会そのものの在りようも問われているのではないだろうか。

## 5.4. 企業内教育

企業内にて行われる教育において、期間については「3ヶ月以内」が有効回答数の約47%であり、次いで「6ヶ月以内」が約21%であった。「1ヶ月以内」「1週間」という短い期間で教育が終了するというのは有効回答数の約24%であった。以下、習熟内容について述べる。

### 5.4.1. 教育内容と習熟度

教育内容と習熟度については、システム開発を「一人で実行できる」との回答は、有効回答数の約32%であり、「指導者の下で実行できる」が約40%であった。企業内で新人として働き始めるにあたり、問題のないレベルまで教育・習熟できていることが窺える。

以下、習熟した科目毎に結果をまとめる。

#### 1. 習熟したオペレーションシステム 回答者：386

70%近くが Windows であり、次いで10%ほどで UNIX、Linux と続く。これは企業の多くが、Windows 環境下において開発を行っているためであると考えられる。

#### 2. 習熟した言語 回答者：418

30%弱がそれぞれ Java と C を学び、次いで10%ほどで C++、COBOL を学んでいる。Java を用いたプログラミング技術が世の中のニーズとして存在し、企業がそれに対応していることがわかる。また、言語の教育にあたり最も一般的と思われる C 言語の教育も重視されている。

#### 3. 習熟したデータベース 回答者：279

Oracle が約60%、次いでそれほど大差がなく HiRDB、DB2/UDB、Access、SQL Server と続いている。最もシェアの高い Oracle に対する教育が、多くの企業で行われていることが窺える。回答者数が OS や言語に比べて減ってはいるものの依然高い数値であるため、データベース技術の必要性が窺える。

4. 習熟したアプリケーションサーバ 回答者：75

30%弱が Tomcat であり、次いで10%強で WebSphere、WebLogic と続いている。その他が30%弱あるため、他にも様々なアプリケーションサーバが使われていることがわかる。教育に用いるにあたり、フリーであり、シェアも高い Tomcat が用いられている。回答者数が少ない。アプリケーションサーバの教育を行っている企業は少ないことが窺える。

5. 習熟した統合開発環境 回答者：77

Eclipse、Visual Studio があわせて50%を超えている。言語の教育を受ける際にこれらの統合開発環境を用いたと考えられる。言語の教育において Java と C の割合が高いことから、Java →eclipse、C(C++)→Visual Studio ということであるのだろう。

6. その他の学習項目

ネットワークが40%強、セキュリティが10%強である。他にプロジェクトマネジメント、オブジェクト指向、システム開発等が挙げられている。ネットワーク、セキュリティ共に、現在の高速通信社会を考えるとこれらの教育を行っているのもわかる。まだ少ないがプロジェクトマネジメント、システム開発は今後重要な教育要素となっていくことが予想される。

## 5.5. 業務を通しての習熟度

実際の案件での開発経験が「ある」と答えた人が約70%の443名であった。主にIT関連企業を対象としたアンケートであり、当然、多くの人が開発という業務を経験していることがわかる。

開発プラットフォームでは、約60%がパソコン、次いでワークステーション、メインフレームと続く。多くの人が個人用のパソコンを用いて開発作業を行っていることであろう。

以下、個別の習熟度について述べる。

1. 習熟度：OS

個人で使えるレベルのOSとしては Windows が最も高く、次いで UNIX、Linux と続く。MAC、TRON などその他のOS を使える人はほとんどいない。開発プラットフォームとして多くの人がパソコンと答えているだけあって、OS も Windows が多く使われているということであろう。

2. 習熟度：言語

C または C++ が最も習熟度が高く、60%弱の人が個人でできるレベルまで習熟している。次ぐ

Java は40%強の人が、同じく個人でできるレベルまで習熟している。C#、COBOL は未習熟である人がほとんどである。指導できるレベルにまで達している人はどの言語であってもそう多くはない。言語として C/C++、Java が使われることが多いため、自然とその習熟度も高まっていると考えられる。指導できるレベルにまで達している人は少なかったが、個人でできることと指導できることとの線引きは難しいため、実際には指導できる人はもっと多いことも考えられる。

### 3. 習熟度：開発技術

リレーショナル型 DB は約50%が個人でできるレベルにまで習熟している。いずれの開発技術要素も個人でできるレベルに達している人が20%いればいい程度で、ほとんどが指導の下でなければできないものとなっている。これに対して指導可能な人の割合は非常に低い。まだまだ新しい技術要素が多いため勉強中である人が多いのだろう。

### 4. 習熟度：プロジェクトマネジメント

経験者の中でも、指導可能な人が少なく、逆に分からないと答えている人が多い。指導の下であればできるという割合は20%強ほどある。経験者自体が少ない。さらにその中でも指導の下であればできる、と答えている人の割合が高い。これより、プロジェクトマネジメントは近年本格的に習熟が要されているものであることがわかる。

## 5.6. 現在の業務

現在の業務の経験年数については、「3年以下」が31.5%、次いで「1年以下」が約24.9%、「5年以下」が17.1%と続く。一つの業務を長く続ける人が少ないことがわかる。これは業務経験年数が経つにつれて違う業務に携わるようになっていくということが考えられる。

ただし、アンケート回答者の潜在的な偏りは考慮する必要があると考えられる。アンケートの配布は、少なくとも入社5年目程度までは一様に配布されている。一方、世代に応じた業務の多忙性の偏りは考えられるので、回答者に若手が多くなったことは想像できる。

一人当たりの回答業務数では、1との回答が約70%を占めた。ほとんどの人が複数の業務に携わっていることはない。

業務分類毎の回答件数から見ると、システム技術者とソフトウェア技術者に大別される。企業の業務内容もシステム開発とソフトウェア開発が業務の主軸であるということだろう。

業務分類毎の経験年数構成では、業務内容によらず、経験年数が5年未満である人がほとんどである。これは、前述の回答者の傾向の影響を考慮する必要がある。

## 5.7. 研修または自分での勉強

以下、分類毎に順次まとめる。

### 1. オペレーションシステム

40%弱でLinux、次いでWindowsが約20%、UNIXが約10%と続く。Linuxの勉強をしている人が多いのは、習熟度が低かったことと、今後企業内でも使用する機会が高まってくると予想されるためであろう。Windows、UNIXに対する習熟度が高かったため、新しく研修や勉強で学習する人は少ない。

### 2. 言語

40%弱でJava、次いでC、C++、C#がそれぞれ約10%で続く。Javaでプログラムを行う機会が今後増えてくることが予想される。企業内教育においても、Javaの割合は高かったことから、企業内教育で不足していた部分や理解不足の部分を勉強しておこうと考えている人が多いとも考えられる。

### 3. データベース

50%強がOracleについて勉強している。シェアが高いデータベースだけに、その影響かと考えられる。PostgreSQLが約10.9%と比較的高いようである。

### 4. 開発ツール

40%強がEclipse、20%強がVisual Studioを勉強している。Javaの勉強と併せてEclipseについても勉強しているものと考えられる。企業内でもJavaを用いた開発時に、Eclipseを使用することになっているとも考えられる。Visual Studioも同様かと考えられる。

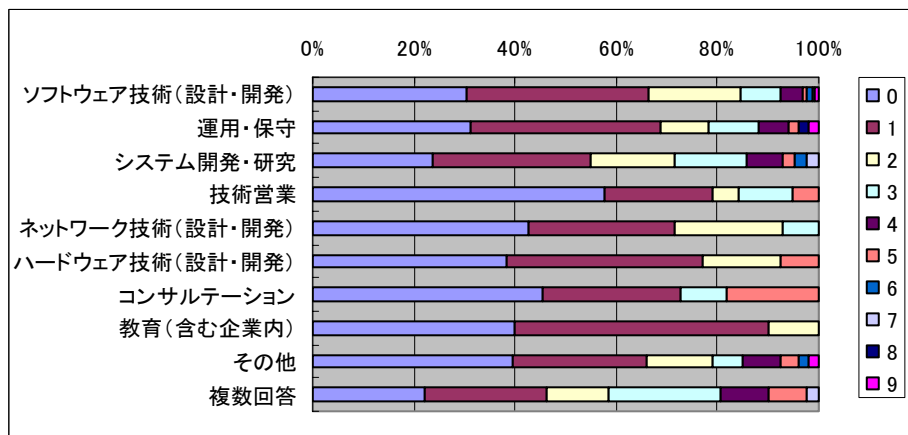
### 5. ネットワーク／セキュリティ

10%強がTCP/IPについて勉強している。その他が非常に多い。その他が非常に多いのは、個別要素技術が多岐に渡ることが影響しているかと考えられる。

## 5.8. 資格

一人が保有する資格数について職種毎に整理すると下表及び下図のようになる（合計値は職種と資格のアンケート質問の両方に回答した人数になる）。なお、回答者に職種の偏りがあるため職種毎の比較は出来ない。

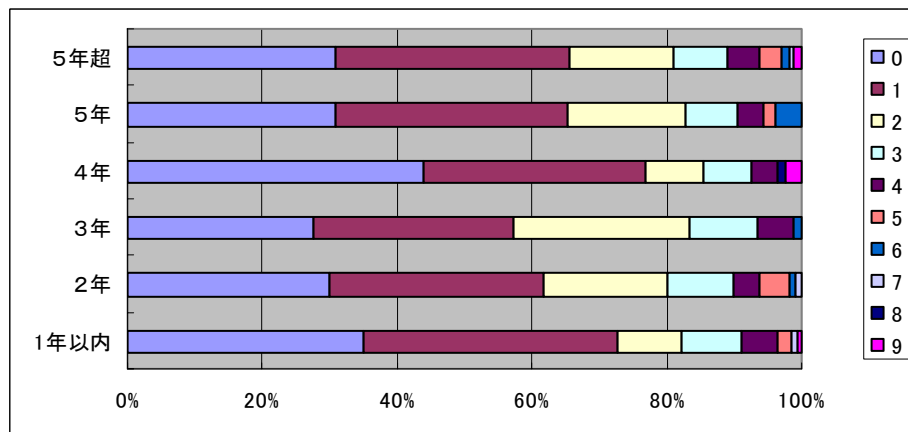
一人当たりの資格保有数	ソフトウェア技術(設計・開発)		運用・保守		システム開発・研究		技術営業		ネットワーク技術(設計・開発)		ハードウェア技術(設計・開発)		コンサルテーション		教育(含む企業内)		その他		複数回答		合計
	人数	割合	人数	割合	人数	割合	人数	割合	人数	割合	人数	割合	人数	割合	人数	割合	人数	割合	人数	割合	
0	111	30.2%	16	31.4%	10	23.8%	11	57.9%	6	42.9%	5	38.5%	5	45.5%	4	40.0%	21	39.6%	9	22.0%	198
1	132	36.0%	19	37.3%	13	31.0%	4	21.1%	4	28.6%	5	38.5%	3	27.3%	5	50.0%	14	26.4%	10	24.4%	209
2	67	18.3%	5	9.8%	7	16.7%	1	5.3%	3	21.4%	2	15.4%			1	10.0%	7	13.2%	5	12.2%	98
3	29	7.9%	5	9.8%	6	14.3%	2	10.5%	1	7.1%			1	9.1%			3	5.7%	9	22.0%	56
4	17	4.6%	3	5.9%	3	7.1%											4	7.5%	4	9.8%	31
5	3	0.8%	1	2.0%	1	2.4%	1	5.3%			1	7.7%	2	18.2%			2	3.8%	3	7.3%	14
6	4	1.1%			1	2.4%											1	1.9%			6
7	1	0.3%			1	2.4%													1	2.4%	3
8			1	2.0%																	1
9	3	0.8%	1	2.0%													1	1.9%			4
合計	367		51		42		19		14		13		11		10		53		41		621



上図表より、「ソフトウェア技術」「運用・保守」で約70%、「システム開発・研究」で約76%が一つ以上の資格を保有していることが分かる。全体的にも高い資格保有率を示している。

同様に、一人が保有する資格数について卒業後年数毎に整理すると下表及び下図のようになる（合計値は職種と資格のアンケート質問の両方に回答した人数になる）。なお、回答者に卒業後経過年数の偏りがあるため卒業後経過年数毎の比較は出来ない。

一人当たりの資格保有数	1年以内		2年		3年		4年		5年		5年超		合計
0	51	34.9%	33	30.0%	25	27.5%	36	43.9%	16	30.8%	50	30.9%	211
1	55	37.7%	35	31.8%	27	29.7%	27	32.9%	18	34.6%	56	34.6%	218
2	14	9.6%	20	18.2%	24	26.4%	7	8.5%	9	17.3%	25	15.4%	99
3	13	8.9%	11	10.0%	9	9.9%	6	7.3%	4	7.7%	13	8.0%	56
4	8	5.5%	4	3.6%	5	5.5%	3	3.7%	2	3.8%	8	4.9%	30
5	3	2.1%	5	4.5%					1	1.9%	5	3.1%	14
6			1	0.9%	1	1.1%			2	3.8%	2	1.2%	6
7	1	0.7%	1	0.9%							1	0.6%	3
8							1	1.2%					1
9	1	0.7%					2	2.4%			2	1.2%	4
合計	146		110		91		82		52		162		643



上図表より、1年目で約65%が一つ以上の資格を保有しており、その後、2、3年目で一人当たり資格保有数が少しずつ増加している。また、4年目以降は若干傾向が異なるようである。

## 5.9. 大学教育への要望

回答者は328名であった。高専・大学・大学院での教育への要望を聞いたところ、皆、真剣に書いて頂いている。今後の詳細な分析が必要であるが、特徴的な点について以下に示す。

演習と講義のバランス：22名、実践から理論・基礎を学ぶ：63名、学問・技術の必要性教育：15名、技術者の講師：8名、インターンシップ：14名、以上で122名であり、実践的な課題から基礎・理論を学べるようにすることを多くが要望している。

海外経験：6名、英語力強化：30名、と企業で実務を遂行するのに英語力を痛感していることが見られる。また、文章力・プレゼンテーション力：11名、コミュニケーション力：10名、グループ開発：8名、などシステム開発者にとって必要となる課題を、是非、学校教育に取組んで欲しい。

最新技術に即した教育：21名、理論・基礎の充実：23名、と言った面から、IT進展の速い業界に働く者として要望されている。実際、教育担当的な立場から、進展を支える基礎・理論を教育する重要さと共に、最新技術とのつながりを常に説明することたろう。

全体観が見えるように、科目の構造化での教育：9名、学問・技術の必要性教育：15名、に対する要望・意見が現れている。

教育・研究共に産学連携の必要性を主張している人が、総論でまとめた中に21名うかがえる。

その他の中に貴重な意見もあり、学校性格別を含め分析を進めていくことが課題である。

## 5.10. 所属する企業

回答者の所属する企業の規模（回答者：628名、約97%）は、約80%が大企業（300人以上）で、業種は約60%がサービス業（システムインテグレータ、ソフトウェア開発）である。「5.1..アンケート回答者」での分析と合わせて考えて、分析はソフトウェア技術者教育の参考になる。

## 5.11. 所属する学会

国内の学会への所属の様相は表[11-a-1,2]に示す通りで、約11.7%の回答者が学会へ所属しており、所属している人において、平均所属数は約1.3である。

一方、国外の学会への所属の様相は表[11-b-1,2]の通りで、所属率は約1.2%、平均所属数は1.3である。所属率で国内と比較すると約10分の1足らずで、国外の研究活動への寄与度の低さが否めないのではないだろうか。

## 謝辞

今回のアンケートによる調査では、膨大で有益なデータが集まりました。特に、自由記述形式部分には皆様の生の声が反映されている結果になっております。業務多忙の中、調査の趣旨にご理解とご賛同を頂き、アンケートにご回答いただいた皆様に、深くお礼申し上げます。また、このような貴重な機会を与えて頂きました経済産業省に対しまして、心より感謝申し上げます。なお、本調査の主旨をご理解いただき、資料提供をいただきました（株）日本経済新聞社、（株）日経BP社に厚く御礼申し上げます。あわせて、今後ともご指導・ご鞭撻の程、よろしくお願い申し上げます。

本報告書の分析では、より多角的な視点で、より深いレベルまでの分析には至らなかったところがあります。今後、さらに本データに基づく分析を行い、大学におけるIT教育の一助となればと考えます。ご協力に再度感謝申し上げます。



2 : 少し役立っている

1 : 役立っていない

	(a) どのように勉強したかを次の区分に従い○でご記入下さい。  4 : 授業で勉強した 3 : 自分で勉強した 2 : 授業と自分で勉強した 1 : 勉強せず	(b) 勉強したときの理解度について次の区分に従い○でご記入下さい。  4 : 十分理解した 3 : 理解した 2 : 少し理解した 1 : 理解できなかった	(c) 勉強したことが役立っているかについて次の区分に従い○でご記入下さい。  4 : 十分役立っている 3 : 役立っている 2 : 少し役立っている 1 : 役立っていない
1 プログラミング入門	4・3・2・1	4・3・2・1	4・3・2・1
2 形式言語とオートマトン	4・3・2・1	4・3・2・1	4・3・2・1
3 プログラミング言語論	4・3・2・1	4・3・2・1	4・3・2・1
1 プログラミング言語体系	4・3・2・1	4・3・2・1	4・3・2・1
2 汎用系 (FORTRAN, Pascal, COBOL, PL-I, etc.)	4・3・2・1	4・3・2・1	4・3・2・1
3 インタプリタ言語 (BASIC, LISP, Prolog, etc.)	4・3・2・1	4・3・2・1	4・3・2・1
4 C言語	4・3・2・1	4・3・2・1	4・3・2・1
5 オブジェクト指向言語 (Java, C++, C#, etc.)	4・3・2・1	4・3・2・1	4・3・2・1
6 スクリプト系言語 (Perl, awk, Python, tcl/tk, etc.)	4・3・2・1	4・3・2・1	4・3・2・1
7 Web系言語 (JavaScript, VBScript, PHP, etc.)	4・3・2・1	4・3・2・1	4・3・2・1
4 計算機科学	4・3・2・1	4・3・2・1	4・3・2・1
1 基礎理論 (情報理論、データ表現、論理演算、最適化問題、他)	4・3・2・1	4・3・2・1	4・3・2・1
2 データ構造 (リスト、スタック、木、ハッシュ、他)	4・3・2・1	4・3・2・1	4・3・2・1
3 アルゴリズム (探索、ソート、再帰、データ圧縮、他)	4・3・2・1	4・3・2・1	4・3・2・1
4 計算機アーキテクチャ (プロセッサ、メモリ、ストレージ、他)	4・3・2・1	4・3・2・1	4・3・2・1
5 情報数理科学	4・3・2・1	4・3・2・1	4・3・2・1
1 多次元データ解析	4・3・2・1	4・3・2・1	4・3・2・1
2 パターン認識	4・3・2・1	4・3・2・1	4・3・2・1
3 マルチメディア (画像、映像、音声、他)	4・3・2・1	4・3・2・1	4・3・2・1
4 AI (エキスパートシステム、他)	4・3・2・1	4・3・2・1	4・3・2・1
5 機械学習 (ニューラルネット、他)	4・3・2・1	4・3・2・1	4・3・2・1
6 最適化手法	4・3・2・1	4・3・2・1	4・3・2・1
8 ファジー	4・3・2・1	4・3・2・1	4・3・2・1
9 カオス	4・3・2・1	4・3・2・1	4・3・2・1
10 エージェント	4・3・2・1	4・3・2・1	4・3・2・1
11 データマイニング	4・3・2・1	4・3・2・1	4・3・2・1
6 ネットワーク工学	4・3・2・1	4・3・2・1	4・3・2・1
1 プロトコルと伝送制御 (TCP/IP等)	4・3・2・1	4・3・2・1	4・3・2・1

2 ネットワークアーキテクチャ	4・3・2・1	4・3・2・1	4・3・2・1
3 伝送制御方式	4・3・2・1	4・3・2・1	4・3・2・1
4 符号化と伝送	4・3・2・1	4・3・2・1	4・3・2・1
5 LANとWAN	4・3・2・1	4・3・2・1	4・3・2・1
6 インターネット応用 (Webアーキテクチャ等)	4・3・2・1	4・3・2・1	4・3・2・1
7 オペレーティングシステム	4・3・2・1	4・3・2・1	4・3・2・1
1 コンパイラの役割と働き	4・3・2・1	4・3・2・1	4・3・2・1
2 UNIX	4・3・2・1	4・3・2・1	4・3・2・1
3 LINUX	4・3・2・1	4・3・2・1	4・3・2・1
4 WINDOWS	4・3・2・1	4・3・2・1	4・3・2・1
5 MAC OS	4・3・2・1	4・3・2・1	4・3・2・1
6 TRON	4・3・2・1	4・3・2・1	4・3・2・1
7 Main Frame OS	4・3・2・1	4・3・2・1	4・3・2・1
8 その他 ( )	4・3・2・1	4・3・2・1	4・3・2・1
8 データベース	4・3・2・1	4・3・2・1	4・3・2・1
1 データベースの原理・理論	4・3・2・1	4・3・2・1	4・3・2・1
2 データの分析と設計方法	4・3・2・1	4・3・2・1	4・3・2・1
3 データ操作 (SQL など)	4・3・2・1	4・3・2・1	4・3・2・1
4 DBMS の機能と特徴	4・3・2・1	4・3・2・1	4・3・2・1
5 分散データベース	4・3・2・1	4・3・2・1	4・3・2・1
6 データベース設計論	4・3・2・1	4・3・2・1	4・3・2・1
7 汎用機のDB (IMS、DB2、他)	4・3・2・1	4・3・2・1	4・3・2・1
8 リレーショナルDB (Oracle、MS SQL Server、DB2/UDB、他)	4・3・2・1	4・3・2・1	4・3・2・1
9 オブジェクト指向DB (ObjectStore、Persistence など)	4・3・2・1	4・3・2・1	4・3・2・1
10 その他 ( )	4・3・2・1	4・3・2・1	4・3・2・1
9 ソフトウェア工学	4・3・2・1	4・3・2・1	4・3・2・1
1 ソフトウェア記述言語 (フローチャート、UML、他)	4・3・2・1	4・3・2・1	4・3・2・1
2 ソフトウェア開発プロセス (ウォーターフォール、XP、他)	4・3・2・1	4・3・2・1	4・3・2・1
3 プロジェクトマネジメント	4・3・2・1	4・3・2・1	4・3・2・1
4 ソフトウェア要求定義	4・3・2・1	4・3・2・1	4・3・2・1
5 ソフトウェア設計方法論	4・3・2・1	4・3・2・1	4・3・2・1
6 ソフトウェア構築方法論	4・3・2・1	4・3・2・1	4・3・2・1
7 ソフトウェア品質	4・3・2・1	4・3・2・1	4・3・2・1
8 ソフトウェアテスト	4・3・2・1	4・3・2・1	4・3・2・1
9 ソフトウェア保守	4・3・2・1	4・3・2・1	4・3・2・1
10 ソフトウェア構成管理	4・3・2・1	4・3・2・1	4・3・2・1
11 ツール及び手法	4・3・2・1	4・3・2・1	4・3・2・1

## 2-2. 印象に残っている授業について

高専、大学の学部・大学院時代に受講した講義・演習で、特に印象に残っている科目があれば、その理由とあわせてご記入下さい。(複数科目でもかまいません)

科目：( )

理由： [ ]

### 3. 教育方法について

高専、大学・大学院での教育方法には、講義、講義と演習（実験）、課題をもとにグループで学習するPBL（プロジェクトベースドラーニング／実際的なシステム開発など）、インターンシップがあり、アルバイト、クラブ活動、などで学ぶ方もあります。その他、卒業論文、修士論文の作成のために色々と勉強したことと思います。以下は、このような教育方法についての質問です。

#### 3-1. 講義・演習について

##### 3-1-1. 授業の方法と満足度について

高専、大学の学部・大学院での、(1)計算機システムの基礎（例：オートマトン、プログラミング基礎等）、(2)ソフトウェア工学関連（例：ソフトウェア開発プロセス、プロジェクトマネジメント等）、(3)情報数理系（例：画像認識、知識データベース等）、について、(a)講義・演習の方法、(b)満足度、をお尋ねします。

各欄の「その他（ ）」には、特に印象深い講義・演習についてご記入をお願いします。

	(a)方法について以下の区分に従い○でご記入下さい。  3：講義のみ 2：講義と演習 1：講義も演習も受 けず	(b)満足度について以下の区分に従い○でご記入下さい。  4：十分満足した 3：満足した 2：やや不満 1：非常に不満
1 計算機システムの基礎	3・2・1	4・3・2・1
1 プログラミング基礎	3・2・1	4・3・2・1
2 計算機概論	3・2・1	4・3・2・1
3 データ構造とアルゴリズム	3・2・1	4・3・2・1
4 その他 ( )	3・2・1	4・3・2・1
2 ソフトウェア工学関連	3・2・1	4・3・2・1
1 ソフトウェア工学	3・2・1	4・3・2・1
2 プロジェクトマネジメント	3・2・1	4・3・2・1
3 システム開発	3・2・1	4・3・2・1
4 その他 ( )	3・2・1	4・3・2・1

3 情報数理系	3・2・1	4・3・2・1
1 知識ベース工学	3・2・1	4・3・2・1
2 生体情報工学	3・2・1	4・3・2・1
3 画像認識工学	3・2・1	4・3・2・1
4 その他 ( )	3・2・1	4・3・2・1
4 ネットワーク系	3・2・1	4・3・2・1
1 通信ネットワーク基礎	3・2・1	4・3・2・1
2 通信方式	3・2・1	4・3・2・1
3 信号処理	3・2・1	4・3・2・1
4 その他 ( )	3・2・1	4・3・2・1

### 3-1-2. 授業の過不足について

現在の技術や業務を通し、高専、大学生当時の講義・演習についての過不足についてお尋ねします。各分野で追加して欲しい科目、内容と不要な科目、内容を思い出してご記入下さい。

	追加して欲しい科目 (内容)	不要な科目 (内容)
1 計算機システムの基礎		
2 ソフトウェア工学関連		
3 情報数理系		
4 ネットワーク系		



(3) テーマ名とその内容を簡単にご記入下さい。

テーマ名：( )

内容： [ ]

テーマ名：( )

内容： [ ]

(3) インターンシップの教育効果へのご意見をお聞かせ下さい。

[ ]

### 3-2-2. 習熟度について

上の質問「3-2-1」で、いずれか又は両方の経験があるとお答えの方で、システムまたはソフトウェア開発関連の場合は次にお答え下さい。

(a) 習熟度について次の区分に従い○でご記入下さい。  4：十分理解し、グループメンバーを助けながら実行 3：理解し自立して実行 2：理解不足で手助けされて実行 1：指示されたままに実行	
---	--

1 内容について		
1 提案書等の書類の書き方	4・3・2・1	
2 システム/ソフトウェア設計方法	4・3・2・1	
3 開発・テスト	4・3・2・1	
4 システム運用・保守	4・3・2・1	
5 オペレーションシステム	4・3・2・1	名称 ( )
6 言語	4・3・2・1	名称 ( )
7 データベース	4・3・2・1	名称 ( )
8 開発ツール	4・3・2・1	名称 ( )
2 最初のPBL・インターンシップについて		

1 開発プラットフォーム（○でご記入下さい）	パソコン・ワークステーション・メインフレーム・その他（ ）
2 規模	（おおよそ 人月）又は（おおよそ 人で ヶ月）
3 開発内容	（テーマ名： ） （概要： ）
4 この過程で学習した科目（データベース、プロジェクトマネジメント）	（科目名： ）
3 二回目のPBL・インターンシップについて（もしあればご記入下さい。）	
1 開発プラットフォーム（○でご記入下さい）	パソコン・ワークステーション・メインフレーム・その他（ ）
2 規模	（おおよそ 人月）又は（おおよそ 人で ヶ月）
3 開発内容	（テーマ名： ） （概要： ）
4 この過程で学習した科目（データベース、プロジェクトマネジメント）	（科目名： ）

### 3-3. 授業以外でのシステム開発経験について

高専・大学時代のクラブ活動、アルバイトなどの活動にてシステムまたはソフトウェア開発経験についてお尋ねします。

#### 3-3-1. 経験について

該当する項目に○でご記入下さい。

- (1) クラブ活動での開発経験：（あり・なし）
- (2) アルバイトでの開発経験：（あり・なし）
- (3) その他での開発経験：（ ）

#### 3-3-2. 習熟度について

上の質問「3-3-1」で、授業以外でのシステムまたはソフトウェア開発経験があるとお答えの方は、それらの経験を通して身につけたことについてお尋ねします。

（複数経験の場合には最も勉強になったケースでお答え下さい）

	(a) 習熟度について 次の区分に従い○ でご記入下さい。  4：指導できる 3：個人でできる 2：指導の下で可能 1：使用できない	
1 開発プラットフォーム	4・3・2・1	パソコン・ワークステーション・メインフレーム・その他( )
2 オペレーションシステム	4・3・2・1	名称：( )
3 言語	4・3・2・1	名称：( )
4 データベース	4・3・2・1	名称：( )
5 開発ツール	4・3・2・1	名称：( )
6 提案書・マニュアル等の書き方	4・3・2・1	
7 規模 (主な経験において)	(おおよそ 人月) 又は (おおよそ 人で ヶ月)	
8 開発内容 (主な経験において)	(テーマ名： ) (概要： )	

### 3-4. 卒論、修論、博士論文について

論文作成は最も効果的な学習の1つだったと思います。そのことに対してお答え下さい。

(1) 課題はどの分野でしたか： 理論的課題、実システムの課題、その他の課題

(2) テーマ名：( )

(3) 内容を簡単に説明してください。

[ ]

(4) 現在の業務で役立っていることはどんなことですか。

[ ]

### 3-5. 学会、論文の経験と効果について

高専・大学時代に学会の論文発表や学術雑誌に論文掲載された経験をお持ちの方にお尋ねします。以下の区分に従いお答え下さい。また、具体的な効果やご意見がありましたらご記入をお願いします。

(1) 学会について (おおよそで結構です)

① 学会への参加 (研究発表していない場合も含む)

1 回数： 回

1-1 国内での学会： 回

1-2 国外での学会： 回

② 学会での発表

1 回数： 回

1-1 国内での学会： 回

1-2 国外での学会： 回

③ 同上の教育効果

該当する項目に○でご記入下さい。

4：非常に効果的である 3：効果的である 2：あまり効果がない 1：まったく効果がない

④ 具体的な教育的な効果やご意見

[ ]

(2) 論文について

① 論文掲載の回数

1 回数： 回

1-1 国内学術誌： 回

1-2 国外学術誌： 回

② 同上の教育効果

該当する項目に○でご記入下さい。

4：非常に効果的である 3：効果的である 2：あまり効果がない 1：まったく効果がない

③ 具体的な効果やご意見

[ ]

## 4. 企業内教育（情報工学関連の教育）について

### 4-1. 期間について

企業内で情報工学関連の教育を受けたことがある場合は、その全累計の期間をお答え下さい。

(1) 期間： 1週間、1ヶ月以内、3ヶ月以内、6ヶ月以内、それ以上

### 4-2. 教育内容と習熟度について

企業内で受けた情報工学関連の教育を通して、身につけたことの習熟度について、以下の区分に従い○でご記入下さい。

- 4：指導できる
- 3：一人で実行できる
- 2：指導者の下で実行できる
- 1：殆ど不可能
- 0：教育受けず

(1) システム開発： 4・3・2・1・0

1 最も習熟したものについて以下の項目毎に名称をご記入下さい。

- 1-1 オペレーションシステムの名称：( )
- 1-2 言語の名称：( )
- 1-3 データベースの名称：( )
- 1-4 アプリケーションサーバの名称（前項 3.6 を参照）：( )
- 1-5 統合開発環境の名称（前項 3.6 を参照）：( )

(2) その他（セキュリティ、ネットワーク、全社システム構成、他）

学習項目：

[ ]

## 5. 業務としてのシステム/ソフトウェア開発を通しての習熟度について

実際の案件でシステムまたはソフトウェア開発経験から身につけたことの習熟度についてお尋ねします。

### 5-1. 経験について

該当する項目に○でご記入下さい。

(1) 実際の案件でシステムまたはソフトウェア開発の経験：(あり・なし)

## 5-2. 習熟度について

上の質問「5-1」で、経験があるとお答えの方は、それらの経験を通して身につけたことについて、次の区分に従い○でご記入下さい。

- 4：グループのメンバーを指導できる程度に習熟
- 3：個人でできる程度に習熟
- 2：指導者の下で可能
- 1：全くわからない

(1) 開発プラットフォーム：該当するものに○でご記入下さい。

パソコン、ワークステーション、メインフレーム、その他（ ）

(2) OS：

- 1 UNIX 4・3・2・1
- 2 LINUX 4・3・2・1
- 3 WINDOWS 4・3・2・1
- 4 MAC 4・3・2・1
- 5 TRON 4・3・2・1
- 6 Main Frame 4・3・2・1
- 6 その他（ ） 4・3・2・1

(3) プログラミング言語：

- 1 C/ C++ 4・3・2・1
- 2 JAVA 4・3・2・1
- 3 C# 4・3・2・1
- 4 COBOL 4・3・2・1
- 5 FORTRAN 4・3・2・1
- 6 その他（ ） 4・3・2・1

(4) 開発技術：

- 1 構造型 DB
- 2 リレーショナル DB 4・3・2・1
- 3 オブジェクト指向 DB 4・3・2・1

- 4 アプリケーションサーバ 4・3・2・1
- 5 統合開発環境/RAD ツール等 4・3・2・1
- 9 その他 ( ) 4・3・2・1

(5)プロジェクトマネジメント：経験のある場合は、下記該当項目で○をご記入下さい。

- 1 統合マネジメント 4・3・2・1
- 2 スコープマネジメント 4・3・2・1
- 3 タイムマネジメント 4・3・2・1
- 4 コストマネジメント 4・3・2・1
- 5 品質マネジメント 4・3・2・1
- 6 コミュニケーションマネジメント 4・3・2・1
- 7 リスクマネジメント 4・3・2・1
- 8 調達マネジメント 4・3・2・1
- 9 組織マネジメント 4・3・2・1

(8)開発時の特記事項（開発時に注意して点、勉強した点等）をご記入下さい。

## 6. 現在の業務について

(1)経験年数 ( )

(2)職種：下記項目で該当する項目番号に○でご記入下さい。

- 1 ユーザー側システム技術者
  - 1-1 プロバイダーへシステム要求
  - 1-2 ユーザーサイドの運用・保守
  - 1-3 ユーザー側でのシステム指導者
  - 1-4 その他 ( )
- 2 プロバイダー側システム技術者
  - 2-1 システムインテグレーター
  - 2-2 プロジェクトマネージャー
  - 2-3 開発者（アプリケーション）
  - 2-4 開発ツール開発
- 3 ソフトウェア開発者

3-1 企画・計画・マーケティング

3-2 研究・開発

3-3 テスト・QA

3-4 サポート・販売支援

4 教育担当

4-1 社内教育

4-2 社外教育

4-3 その他 ( )

5 その他の情報システム関連

5-1 システム関連コンサルタント

5-2 エバンジェリスト

5-3 学術雑誌等の記者

5-4 その他 ( )

5 システムとは関係なし

職種 ( )

## 7. 現在、研修または自分で勉強していることについて

現在勉強中である場合に、下記の該当する項目についてそれぞれご記入下さい。

(1) オペレーションシステム : ( )

(2) 言語 : ( )

(3) データベース : ( )

(4) 開発ツール : ( )

(5) ネットワーク/セキュリティ : ( )

(6) その他 : ( )

## 8. 資格（取得済み・目指している資格）

現在、取得済みの資格と目指している資格をご記入下さい。

(1) 取得済み資格	
------------	--

(2) 目指している資格	
--------------	--

## 9. 要望

国際化する社会の中で、大学教育に対して国際的な技術者、研究者育成への要請がより高く、強くなってきています。このような状況下で、現在、ご活躍の皆様の高専や大学の学部・大学院への教育内容、方法などについての要望を自由にご記入下さい。

## 10. 所属する企業について

現在、所属している企業についてお尋ねします。

(1) 規模：該当する項目番号に○でご記入下さい。

- 1：大企業（300人以上）
- 2：中企業（100～299人）
- 3：小企業（10～99人）
- 4：SOHO（1～9人）

(2) 職業：該当する項目項目に○でご記入下さい。



---

禁無断転載

大学等におけるIT教育実態調査報告書  
(情報系学科卒業生の活動状況調査事業)

平成16年3月

IT教育実態調査プロジェクト  
プロジェクトリーダー: 嘉数侑昇(教授)  
〒060-8628 札幌市北区北13条西8丁目  
北海道大学大学院 工学研究科  
システム情報工学専攻  
複雑系工学講座 自律系工学分野  
株式会社テクノフェイス  
〒060-0001 札幌市中央区北1条西3丁目3  
札幌三和ビル9階

---