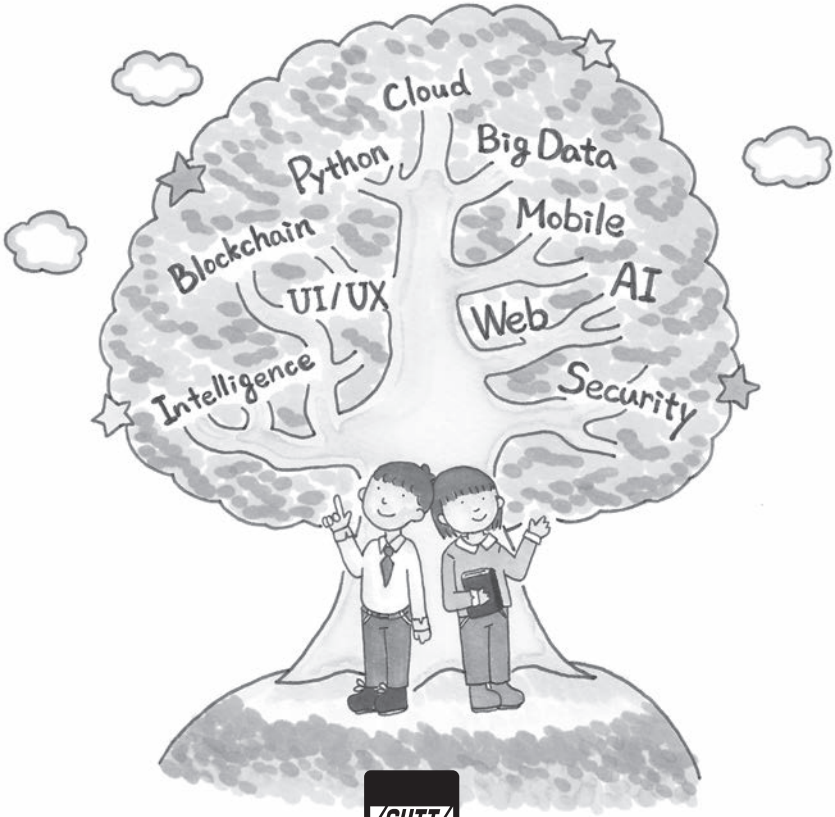
情報処理エンジニア 職業ガイド

プログラマ・ITエンジニア・SEのための キャリアデザイン

豊沢 聡／**大間 哲**◉共著　**加藤ゆきの**◉イラスト



CUTT
カットシステム

本書で取り上げられているシステム名／製品名は、一般に開発各社の登録商標／商品名です。本書では、™ および ® マークは明記していません。本書に掲載されている団体／商品に対して、その商標権を侵害する意図は一切ありません。本書で紹介している URL や各サイトの内容は変更される場合があります。

はじめに

コンピュータにかかわりのある仕事をしたいと思っている人は多いでしょう。

ソーシャルネットワークで友だちとつながったり、オンラインショッピングをしたり、映像や電子書籍などのデジタルコンテンツを楽しんだりと、その技術が身近なのが理由でしょう。また、政府や産業界やマスコミがITだAIだIPだとよくわからない略語を交えて騒いでいるので、旬な話題と思ったからかもしれません。将来なりたいものといえば野球選手やお花屋さんだった小学生たちも、今ではゲーム制作者やユーチューバーを挙げるようになっています。

しかし、具体的にどんな仕事かとなると、プログラマやシステムエンジニアくらいしか出てこないのではないでしょうか。仕事内容も、ぼんやりと想像できるくらいです。就職してその先のキャリアにいたっては見当もつきません。

本書は、コンピュータソフトウェアエンジニアリングにかかわる約120種の職を具体的な求人例を引きながら紹介したものです。各職では、職務内容と必要な技能を示したので、何をするのか、どのような知識や経験が求められているのかがわかるようになっています。ソフトウェア業界に進もうとは思うものの仕事の具体的なイメージのわかない学生さん、あるいは次のステップを模索しているエンジニアの皆さんなら、各職の特徴を他と対照しながら調べるハンドブックとして利用できます。教育機関で就職支援をされる教職員、就職センターやキャリアコンサルタントの方々なら、それぞれの職で必要とされる技能や経歴を調べるのに活用できます。米国を中心とした英語圏の最近の求人例を参考にしているので、先駆的な人事制度やポジションを考えているコンピュータ関連企業の人事担当の方にも参考になると思います。

とはいうものの、残念ながら、これでソフトウェアエンジニア職すべてをカバーできたわけではありません。もちろん、労働統計や求人サイトの見出しに登場する一般的な職は積極的に採用していますが、取捨選択には筆者らのこれまでの業界経験が作用しており、バイアスがかかっていないとはいえません。しかも、ポピュラーでもなさそうなのに、筆者らがこんな仕事ならやってみたいと思っただけで掲載したものもあります。読者のご宥恕を乞う次第です。

本書の企画は、コンピュータ業界に長く身を置いている筆者らでも聞いたこともない、周りにもいない職がたくさんあることに驚いたところからスタートしています。

「13 歳のハローワーク」もあります[※1]。理工系、特にコンピュータ系の仕事が手薄に思え、その分野の適切な紹介が必要だと感じたのも、モチベーションのひとつです。もっとも、実際に着手すると、バランスよく主要な職を列挙するのがいかに困難かわかりました。しかも、「13 歳のハローワーク 公式サイト」に掲載されている 1,000 ちょっと（執筆時点の数）の職業のうち、コンピュータ技術関連が約 3% を占めていることも遅まきながら知りました。3% というと少なさそうですが、日本の労働者人口に対する IT 系従事者の割合が 1.3% 程度なので、他の職よりも多く紹介されているといってよいでしょう[※2]。

本書の大部分は 120 種の職を並べている便欄のようなものですが、末尾の章では、これからのキャリアを設計する上で便利な方法を説明しました。自分に合う生き方、そしてそれを実現する職を探すにあたって、本書を羅針盤として活用していただければ幸いです。

著者一同

※1　村上龍，はまのゆか：「13 歳のハローワーク」，幻冬舎.

※2　統計は 2017 年度のもの。労働者人口は総務省統計局からで 6,750 万人、IT 人材数は経済産業省からで約 90 万人として計算しています。もっとも、値は定義や調査方法で変わってくるので、1.3% は目安程度です。

● 本書の構成

本書掲載のコンピュータソフトウェアエンジニア職は、コンピュータ関連学術団体の ACM と IEEE が示す大学カリキュラムの分類に従ってそれぞれの分野別の章に収録しました。

- 第 1 章はソフトウェア工学（Software Engineering）にかかわる職です。要求分析、設計、開発、テスト、品質管理、運用、保守といったソフトウェア開発工程を内包する開発職で、いわゆるソフトウェアエンジニアです。プログラマもこのカテゴリーに含まれますが、より広い範囲をカバーしています。
- 第 2 章は情報システム（Information Systems）の職です。ソフトウェアやハードウェアを直接には開発しませんが、これらの要素を組み合わせてコンピュータシステムを構築する職で、システムエンジニアです。ただし、この章でも説明しますが、日本で SE と総称される職とはニュアンスが異なります。
- 第 3 章は情報技術（Information Technology）の職です。第 1 章、第 2 章の職がモノを作り上げることに力点を置いているのに対し、このカテゴリーはそうして作られたコンピュータを効果的に利用することが主目的です。略すと IT で、コンピュータ関連ならなんでも総称する略語として日本では扱われがちですが、本書では英語圏の定義に従ってより限定的にこの語を用います。コンピュータ関連ならなんでも包括する職業カテゴリーは、本書では「コンピュータ関連職」と呼びます。
- 第 4 章はコンピュータ科学（Computer Science）の職です。第 1 章から第 3 章の活動のもととなる基盤知識を生み出す科学的な活動に従事します。工学に対する物理、化学、数学といった自然科学分野に相当します。
- 第 5 章はその他の職です。ここには、第 1 章から第 4 章にかかわる活動の管理やサポートなど、技術そのものが仕事の中心ではないものの、技術にはある程度明るいことが求められる職を収録しています。たとえば、開発工程を管理するプロジェクトマネージャ、技術文書を書くテクニカルライター、技術を教えるインストラクターなどです。

各章の冒頭では、その分野の説明、分野に共通する職の概要、そしてキャリアパスにかかわるトピックを示しました。いずれの章も職種別に節で分けていますが、

おおむね一般的な職種から専門性の高い職種の順に並べてあります。また、それぞれの節では、初心者レベルから難易度が高い、あるいは専門性の強い職へとレベルアップしていく順にリストアップしています。

最後の第 6 章では、キャリアデザインを取り上げます。まずキャリアデザインの考え方から始め、実際にキャリアデザインを行う方法を具体例を挙げて説明します。続いては、コンピュータ関連の仕事環境におけるキャリアパスの描き方を、第 1 ～ 5 章の情報を用いて考えます。

最初から通読する必要はありません。必要に応じて、必要な箇所だけ読んでくれてかまいません。

● 本書掲載の職について

前述の章立てからわかるように、本書で取り上げているのはコンピュータソフトウェア系のエンジニアあるいはその周辺の職だけです。コンピュータを活用するものの技術的ではない職、たとえばゲームストーリーライター、ユーチューバー、CG アーティストなどは取り上げていません。また、紙面の都合からソフトウェア系に限っているので、回路設計や半導体製造などハードウェアや素材開発といったコンピュータ工学系も割愛しています。ハードウェア指向の方々には、ソフトウェア屋の筆者らの怠慢をお詫び申し上げます。

参照した求人サイト、大学、労働統計はいずれも米国を主とした英語圏のものです。

それぞれの職で参照している求人例は、これらの情報からエッセンスを抽出して数段落でまとめたものです。1 段落目が職務内容、2 段落目が要求される技術および経験です。求められる認証資格は各項目では名称くらいにとどめ、まとめて付録 A に説明を記載しました。その他の要求条件は、特記すべきものがあれば 3 段落目に加えました。技術そのものを避けることはできませんが、専門家を相手にした専門的な記述はわかりやすいように言い換えています。そのため、技術的には正確ではない記述もありますが、ご容赦ください。

求人には正社員（長期雇用）やパートタイムなど雇用形態が記載されていますが、これらは本文から省いています。同様に、雇用条件や福利厚生制度などの特典も省きました。参考までに、それらはまとめてそれぞれ付録 B と付録 C に掲載しました。

日本の求人例は含まれていませんが、これは意図的なものです。

日本の求人も、最近では中途採用のものは職務や技能の記述が充実してきているので、職の具体的イメージはつかみやすくなってきました。しかし、ひとつの求人に複数の職やポジションを盛り込んでいるためか、求人元の望みがあいまいです。たとえば、とある大手総合エレクトロニクスメーカーの中途採用向けのソフトウェア開発者の求人に列挙されている技術は「AI、IoT、ブロックチェーン、モバイル、クラウド、ディープラーニング、ビッグデータ、アナリティックス、Elasticsearch および Kibana、Mattermost など主としてオープンソース系の最先端技術」で、ターゲットが絞れているとはいえません。また、ポジションや待遇も不明瞭です。初級なのか上級なのかといった社内での地位（イコール給与水準）および要求される技術レベル、単体実装担当なのか全体を掌握するアーキテクトなのかという仕事の範囲がわからないと、自分の現在の立ち位置と職をマッチできませんし、ましてやキャリアパスを描くのは困難です。加えて、日本の労働市場は年齢制限がある、内部登用が多い、年季を経ると技術から遠ざかる傾向が現実にはあり、上級レベルのエンジニア職の事例が少ないこともあります。

これは、日本では会社全体として新卒一括採用をし、のちに部署に割り振る手順が主流のためでしょう。また、どこの部署でも通用するジェネラリストが好まれるため、広く浅くとったほうが融通がきくという思惑もあるでしょう。

これに対し、英語圏の求人は求める職務をきわめて限定するため、その職の内容や要求レベルを読み取るのに適しています。また、エントリーからエキスパートまで幅広いレベルの職が求人されており、キャリアパスを考える上でも参考になります。

さらに、本書に示したどちらかといえば新しめな職種や職名も、技術そのものと同時に日本にも導入されることが見込まれるので、将来を見込んだ職を盛り込めるというメリットもあります。昨今では就職活動もグローバル化しており、同じ会社であっても日本と海外では採用基準が異なることもまれではありません。そうしたグローバルな環境を志向する方にも、本書の情報は有益だと思います。

1
おしごと きめたー？
うーん
わたしは まだ いろいろ なれんそで…
WORK
2
そうだ！！
IT関係が いいな！ かっこいいし!!
グッ
パソコン とくい だし！
3
なるほど〜
具体的に どんなおしごと??
!?
あ…えっと… それはこれから…
ITか〜！ おもしろそう！
WORK
4
ITのおしごとに 興味がある キミたち!!!
5
ぱぱぱぱ——ん
IT系には どんなおしごとがあるのか、 ぼくたちがおしえてあげるよ!!!
あなたの 応援隊!!

目次

はじめに iii

1 ソフトウェア工学 1

ソフトウェアエンジニア 2
職業としてのソフトウェアエンジニア 2

1.1 ソフトウェア開発者（一般） 4

コラム 「職級」 5

ソフトウェア開発インターン 6
ソフトウェア開発者・初級 7
ソフトウェア開発者・中級 9
ソフトウェア開発者・上級 10
ソフトウェアアーキテクト 13

1.2 ソフトウェア開発者（対象別） 15

Python 開発者 16
C/C++ 開発者 17
COBOL・メインフレーム開発者 19

コラム 「プログラミング言語トップ 10」 20

プログラマアナリスト 22
組み込みソフトウェアプログラマ 23
ゲームプレイプログラマ 25
GPU プログラマ 26
IoT ソフトウェアエンジニア 28
ブロックチェーンエンジニア 30
API 開発者 31
コンパイラ開発者 33

1.3 ウェブ開発者 34

フロントエンド開発者 36

バックエンド開発者 37
フルスタック開発者 39

1.4 デジタル信号処理エンジニア 40

オーディオ信号処理エンジニア 41
画像処理エンジニア 42
ソフトウェア信号処理エンジニア 45
コラム **「国籍条項」 46**
自動運転エンジニア 46

1.5 グラフィックスエンジニア 48

3DCG エンジニア 49
VR エンジニア 50
テクニカルアーティスト 51

1.6 開発工程管理 52

ソフトウェアビルドエンジニア 53
ソフトウェアテストエンジニア 54
リリース管理エンジニア 55

1.7 その他 57

UI/UX デザイナー 57
国際化エンジニア 59

2 情報システム 63

情報システムエンジニア 64
職業としてのシステムエンジニア 66

2.1 営業エンジニア 66

プリセールスエンジニア 68
ポストセールスエンジニア 69
営業エンジニア 71
コラム **「新人レベルに最もポピュラーな職種」 73**
アカウント営業エンジニア 74
営業ソリューションアーキテクト 75

2.2 フィールドエンジニア 77
フィールドエンジニア 78
コラム **「身体負担のある職業」 80**
システム設置エンジニア 80
2.3 データベースエンジニア 81
データベースエンジニア 83
NoSQL データベースエンジニア 84
SQL プログラマ 86
ETL プログラマ 87
データアーキテクト 89
コラム **「一番儲かるコンピュータエンジニア職は」 91**
データモデラー 92
ビジネスインテリジェンスエンジニア 94
2.4 エンタープライズソフトウェアエンジニア 96
CRM エンジニア 97
ERP エンジニア 99
EHR エンジニア 100
2.5 アナリスト 102
データアナリスト 103
ビジネスアナリスト 104
ワークフローアナリスト 107
システムアナリスト 108
2.6 コンサルタント 109
営業コンサルタント 111
対外コンサルタント 113
情報セキュリティコンサルタント 114
2.7 スペシャリスト 116
ネットワークシステムエンジニア 117
クラウドエンジニア 120
サイト信頼性エンジニア 122
CAE エンジニア 123
セキュリティエンジニア 125
コラム **「セキュリティとゲーマー」 126**

RPA スペシャリスト 127
VFX パイプラインエンジニア 128

3 情報技術 131

IT エンジニア 132
職業としての IT エンジニア 133

3.1 オペレータ 134

IT オペレータ 134
ネットワークオペレータ 137
IT システム導入オペレータ 139
IT 資産管理オペレータ 141
IT オペレーションアナリスト 143

3.2 システム管理者 145

システム管理者 146
仮想化システム管理者 148
データベース管理者 149
ネットワーク管理者 151
ウェブ管理者 152

3.3 技術サポートエンジニア 154

顧客サービス担当 157
技術サポートエンジニア・レベル 1 158
技術サポートエンジニア・レベル 2 160
技術サポートエンジニア・レベル 3 161
コラム **「上級サポート職と開発職」 163**
IoT サポートエンジニア 163
DevOps エンジニア 165

3.4 その他 167

技術助監督 167
IT コーディネータ 168

4 コンピュータ科学 171

コンピュータサイエンティスト 172
職業としてのコンピュータサイエンティスト 174

4.1 基礎系 174

コンピュータサイエンティスト 175
人工知能サイエンティスト 177
プログラミング言語サイエンティスト 179

4.2 応用系 181

コラム 「博士（はくし）が 100 にんいるむら」 ---------- 182
人工知能サイエンティスト（応用） 182
自然言語処理サイエンティスト 183
画像処理サイエンティスト 185
コンピュータグラフィックスサイエンティスト 187
データサイエンティスト 189
通信システムサイエンティスト 191
ユーザインタフェースサイエンティスト 192

4.3 複合系 194

計算化学サイエンティスト 194
生物計算学サイエンティスト 195
社会科学系コンピュータサイエンティスト 196

5 その他 199

5.1 技術マネージャ 199

プロジェクトマネージャ 200
プロダクトマネージャ 203
プログラムマネージャ 204
IT 変更管理マネージャ 206
最高情報責任者 208

コラム 「CIO になるための 5 つのステップ」 ---------- 209

5.2 文書・デザイン 210

テクニカルライター 210

ローカライゼーションスペシャリスト……212
翻訳家……214
コラム「言語の熟達レベル」……215
UI/UX コピーライター……216
SEO スペシャリスト……218
テクニカルアーティスト……219

5.3 教育……220
教材開発者……222
インストラクター……223

5.4 法務……225
知的財産パラリーガル……225
特許分類付与担当……227
カウンセル……228
IT コンプライアンスアナリスト……229
データ保護責任者……232
デジタル法科学者……234
AML スペシャリスト……235

5.5 その他……237
ゲーム店店員……237
通信料金分析担当……237
データ入力事務員……239

6 キャリアデザインの実践……241

6.1 キャリアデザインとは……242

6.2 キャリアデザインの実践……243
ライフラインチャート（過去）……244
バリューカード……248
ライフラインチャート（未来）……252
Want-Can-Must 図……253
今後のキャリアデザイン……257

6.3 コンピュータ関連職のキャリアパス 258
一般的なキャリアパス 259
キャリアパス上の障害と考え方 263
キャリアパス再訪 264
6.4 キャリア相談 264
6.5 おわりに 267

付 録 269

付録 A 認定資格 269
付録 B 雇用形態 276
問題が生じたら 281
付録 C 特典 283
付録 D キャリアデザインツールキット 290
ライフラインチャート 290
バリューカード 291
Want-Can-Must 図 292

索 引 293

1

ソフトウェア工学

ソフトウェア工学（Software Engineering）はソフトウェア（プログラム）の要求分析、設計、開発、テスト、品質管理、運用、保守にかかわる工学分野です。「工学」とあるように、コンピュータ科学（第 4 章）で得られた基礎知識を応用し、経済性、実用性といった制約のもと、目標を達成するベストなソフトウェアを生み出すことを目的としています。

学術分野としてのソフトウェア工学には次のトピックが含まれます。

- 要求分析と仕様化
- 開発手法
- ソフトウェアモデル
- ソフトウェア設計（アーキテクチャ、ハードウェア、データモデルとその実装、トレードオフ解析、ヒューマンインタフェースを含む）
- プロセス解析
- ソフトウェアセキュリティ
- テスト手法
- 品質管理
- プロジェクト管理

ソフトウェア工学とコンピュータ科学の違いは、あまり明確ではありません。モノづくりの立場からコンピュータの新しい知識を生み出すエンジニアもいれば、実

用的なソフトウェアを書くサイエンティストもいるからです。あえていえば、コンピュータ科学は理論的側面を扱い、ソフトウェア工学はその成果物を利用するという分けかたが一般的です。たとえば、プログラム言語そのものを設計するのはコンピュータサイエンティストで、それを使って実際に動作するアプリケーションを構築するのがソフトウェア開発者です。

● ソフトウェアエンジニア

ソフトウェアエンジニア（Software Engineer）はソフトウェア工学の知識と技術を用いてソフトウェアを生み出していく職です。一般にプログラマと呼ばれる職はこれに含まれます。

もっとも、ソフトウェアエンジニアはプログラムを書くだけの存在ではありません。ソフトウェアの設計にあたっては、ユーザの顕在的潜在的なニーズを満たし、実用性を高めるようにユーザの要求を分析します。そして、要求を技術的な言葉で書き起こした仕様を作成します。設計では、最終的な製品からバグを減らし、信頼性を高くし、メンテナンスも容易になるように考察もします。コストも重要な観点のひとつです。ひとつのプログラムは、それぞれ個別に作成する多数のモジュールや外部から導入するライブラリを組み立てることで構築されますが、これらの要素のそれぞれの関係や組み立てかたを決めるのも仕事のうちです。ソフトウェアができあがったら、もともとのユーザニーズを満たしているか、バグがないか、性能上の問題はないか、安定しているかなど品質を評価します。また、こうした活動を担当する多くのエンジニアを集めたソフトウェア開発プロジェクトを円滑に実行するべく、スケジュールやリソース（人員や資材）を管理します。

● 職業としてのソフトウェアエンジニア

ソフトウェア開発にかかわる活動は上記のように多岐にわたりますが、ひとりのエンジニアがすべてをカバーすることはまずありません。プログラマはプログラムを書くこと、品質管理担当はテスト項目を作成してそれを実施することにそれぞれ専念します。したがって、それぞれの担当別に固有の職が用意されています。「ソフトウェアエンジニア」のようなおおざっぱなポジションもないわけではないですが、それは新人向けであったり、ソフトウェア開発がその組織の主な業務ではない補助的な役割であることがほとんどです。

もちろん、どの活動もソフトウェア開発には必須なものなので、専門家でなくともその要領をある程度はわきまえていなければなりません。少人数で開発をする小規模なプロジェクトや組織では、専門家ほどうまくはできなくとも、ひとりでいくつもの活動を担当せざるをえないこともあるからです。

ソフトウェアエンジニアのキャリアパスも他の職業同様、多様です。

まず、狭い守備範囲から始め、経験を積むに従い、徐々に大局的な考え方が要求される戦略的なポジションへ上がるという道筋があります。たとえば、コンピュータ工学あるいはソフトウェア工学を大学で修めたのち、ひとつのモジュールを仕様どおりに作成するプログラマから始めます。その後は、自分で仕様を書くような上級の開発者になり、その先は、製品全体の構造を設計するソフトウェアアーキテクトへと進みます。また、その過程で扱える言語やツールを増やしていくことで、カバーできる要求や実装も広げていきます。

ソフトウェアが利用される分野の知識を深めていくというパスもあります。仕様も書ける上級ソフトウェア開発者になれば、たとえばこれまでの銀行システムソフトウェア開発の経験から金融系の知識が蓄えられています。医療情報処理ソフトウェアなら医療関係の知識です。そこから、金融業界固有の商品や業務プロセス（手続き）、要求条件、法令などの知識をベースに、金融システム専門ソフトウェア開発者へと発展します。

経験を積むにつれ、開発手法、モデリング、システム解析、要求分析、品質管理といった開発を支援する各種の専門知識も身についてきます。年季を経てそれら専門職へと進むのも、ソフトウェアエンジニアによくあるキャリアパスです。もちろん、これらの領域を専攻する大学のコースもあり、新卒からその道に飛び込むこともできます。逆に、品質管理からソフトウェア開発に戻ってくることもあるので、そちらが最終到達地点というわけではありません。

ソフトウェア開発ではなく、たとえばシステム工学（第２章）分野のビジネス要求分析職、あるいは情報技術（第３章）分野のサポート職へと転身することも当然あります。プロジェクトを管理するプロジェクトマネージャ、あるいはソフトウェア製品の開発戦略を担うプロダクトマネージャのような、コンピュータ技術にはじかには触れない技術管理職へと進むこともあるでしょう（第５章）。ただし、システムエンジニアやITエンジニアからソフトウェア開発に進む逆の道は、日常的にプログラミングをしていなければ、やや険しいです。そういう意味では、ソフト

ウェア開発職は（技術的な職という範囲内では）他よりもやや自由度が高いといえます。

ソフトウェアはほとんどすべての企業活動に必須なため、ソフトウェアそのものの製造販売会社以外でもソフトウェア開発者が求められています。大規模なデータを扱う官公庁、軍や警察、金融、教育などはお仕着せの製品だけでなく、自分たちの目的に合ったソフトウェアの自社開発やカスタマイズも多いため、求人も活発です。しかし、分野によって要求される技術レベルが異なる点には注意が必要です。ソフトウェアや情報システムと本来的にそれほど関係のない産業や会社では、簡単な実装くらいしか要求されず、スキルの向上が望めない可能性もあります。かといって、新卒からいきなり Google や Facebook の開発者になるのは難しいのも確かです。自分のスキルの方向やそのための経験を積む機会があるかを考えながら、就職先を選んでいかなければなりません。

1.1 ソフトウェア開発者（一般）

プログラミング言語を用いてソフトウェアを書くのがソフトウェア開発者（Software Developer）です。いわゆるプログラマ（Programmer）です。

アプリケーション開発者（Application Developer）とも呼ばれます。アプリケーションは特定の目的のために特定のユーザに供するソフトウェアという意味なので、ソフトウェア開発者の大半はアプリケーション開発者だからです。もちろんオペレーティングシステム、デバイスドライバ、ネットワークプロトコル処理などエンドユーザが直接タッチしない基盤的なソフトウェアもありますが、この手の求人は数が限られます。

ソフトウェア開発者の職は主として技能レベル別と開発対象別にわけることができます。本節では前者を、次節では後者を扱います。技能レベルで最もわかりやすいのは数値でのレベル分けです。たとえば、ソフトウェア開発者I、II、IIIです。ジュニアやシニアという修飾で示すこともあります（コラム参照）。レベルは業務経験年数と密接に関係しているため、求人では経験何年以上と明記されるケースがほとんどです。

レベルは、その人の開発工程における担当範囲の広さも示しています。初級レベ

ルは開発だけ、中級は設計やテストも担当、上級は全工程を掌握する、といった塩梅です。ソフトウェア機能の担当範囲も示しています。初級は一部のモジュールだけに専念し、レベルが上がるにつれて把握する範囲が広がります。上級ともなると、ソフトウェア単体だけではなく、システム全体の整合性も考慮します。扱う言語や開発環境のレパートリーもレベルが上がるにつれて増えてきます。反対に、ソフトウェアの応用分野は金融データ処理専門など、レベルが上がるにつれ範囲が狭く深くなっていきます。

プログラミング言語の盛衰は激しく、若いころにポピュラーだった言語が壮年時にも需要があるかはわかりません。その昔は COBOL や Fortran がメインでしたが、今では求人を見つけるのも困難です。Perl も 20 世紀にはよく使われていましたが、今では Python などの言語にとって代わられています。C/C++ や Java など（コンピュータ言語にしては）長い歴史を持つ定番もありますが、昔のように汎用的に使われるのではなく、用途限定になっています。そのため、プログラマは常にアンテナを張り巡らし、転職時に有利な言語を学んでいかなければなりません。

職級

ジュニアやシニアくらいならどちらが上か下かは判断がつきますが、見当がつきにくい修飾もあります。レベルの低い順に次に示します。

- Intern（インターン）
- Apprentice（見習い）
- Assistant（助手、アシスタント）
- Associate（準職員）
- Member（正職員）
- Major（上級職員）
- Principle（特別職員）
- Leader（チームリーダー）

会社や業界によって上下が反転することもありますが、だいたいこんなものです。ちなみに、日本の官公庁やそれに類する組織では「5 等級」など数値を用いており、ここだけはデジタル化が進んでいます。もっとも、ドストエフスキーの小説にも「第十等」などと出てきますので、帝政ロシアと同レベルと考えるべきかもしれません。

● ソフトウェア開発インターン（Software Developer, Intern）

業務経験がほとんど必要とされないインターンレベルは、ソフトウェア全体のごく一部（モジュール）のプログラミングから始めます。モジュールの設計を任されることもないわけではありませんが、たいていは上級エンジニアの設計どおりにコードを書くのが仕事です。ソフトウェアテストもやりますが、自分の書いたコードが適切に動作するか程度の単体テストが主体です。その過程でバグ取りなど問題解決手法、開発プロセスなどを学んでいきます。

損保会社の求人例を次に示します。

Windows あるいはウェブで動作するアプリケーションを作成するソフトウェア開発インターンを求めています。業務には既存のソフトウェアやデータベースの更新、ソフトウェアテストの実施、報告された問題の調査、不具合のあるソフトウェアの修正も含まれます。このインターンを通じ、SQL Server、Visual Studio、Visual Basic、C#、ASP.Net といった Microsoft の開発環境、レポート作成ツール（Crystal Reports）を学ぶことができます。より進んだ学生なら、JavaScript、jQuery、HTML、CSS、C#.NET などのプログラミング技術や REST をはじめとするウェブ通信のコンセプトも学べます。

対象はコンピュータ科学あるいは数学専攻の 2 ～ 4 年の学部生です。Microsoft Windows と Office、そしてキーボードが使えれば歓迎します。

インターンはご存知のように、学生が夏休みなどの期間に就業経験を積むためのプログラムです（付録 B 参照）。

この求人にはソフトウェアにどんな機能を加えるかを決める要件分析や設計は含まれておらず、テストもつけたし程度です。言語が数点並んでいますが、これから扱うものであり、現在有するスキルの指定ではありません。経験があると有利でしょうが、そこまで厳密ではないのが一般的なインターンの特徴です。応用分野はウェブ以外は明示されていないので、アプリケーションの適用分野の知識がなくても対応できる範囲です（ウェブ開発の専門職は 1.3 節で扱います）。つまり、大学あるいは専門学校のソフトウェア作成プロジェクトを経験していれば、おおむね対応できる技能レベルです。

キーボードが要件に入っているのは、タッチスクリーンしか使わない世代が増えたことを反映しているのでしょう。プログラミングは、さすがにタッチだけでは難しいです。

リストから明らかなように、扱う技術が Microsoft 系に偏っています。しかし、これはこの求人に固有というわけではありません。開発環境や関連するシステム要素は、連携性がよいことから、このように単一のブランドに集中しがちなのです。そのため、一社だけで経験を積んでいると、同じ職種であっても、他環境への転職時に蓄積してきた経験が生かせなくなることもあります。

今度は Unix 系に偏った求人例を、自動車部品および電気製品を製造販売する会社のものから示します。

音声処理技術を研究開発するサイエンティストを支援するインターンを求めています。業務はプロトタイプの設計、開発、テストです。

応募者はコンピュータ科学あるいはコンピュータ工学専攻の 3、4 年の学部生でなければなりません。また、Linux、Eclipse 開発環境、C/C++ プログラミングの技能、そしてデータ構造に関する知識が求められます。

研究者は新しい処理方法を考案すると、その方式が適切か、思ったとおりの性能を発揮するかを確認するため、プログラムを試作します。これがプロトタイプです。研究者本人が自力で実装してもよいのですが、アシスタントに依頼することで作業効率を上げます。そのため、このインターン職は研究者の思い描く高度なアルゴリズムを理解できるだけの素地を必要とする、かなりレベルの高いものとなっています。また、C/C++ が対象と、利用言語でも難易度は上がっています。

音声処理の確立した技術を応用するエンジニアリング職は 1.4 節、未解決な課題に取り組むサイエンティスト職は 4.2 節をそれぞれ参照してください。

● ソフトウェア開発者・初級（Software Developer, Level 1）

新卒あるいは数年程度の実務経験のある人を対象とした、エントリーレベルのソフトウェアエンジニア職です。単に「プログラマ」の場合、おおむねこのレベルに含まれます。

初級ソフトウェア開発者は大きなプログラムの中の一部分（関数やクラスなど）だけを定められた仕様に従って作成します。料理人でいえば、ジャガイモの皮をむいたりお皿を温めたりすることで、より上位の料理人の調理を手助けする新米コックです。でも、決して雑用係ではありません。新米でもプロです。技能的には特定の言語の用法をすべてではないにしてもだいぶ知っており、小さなプログラムなら自力で書けるレベルが必要です。ソフトウェア開発に必要な要件定義やテストなどの工程もある程度は任され始めます。

ケーブルテレビやブロードバンド通信用の機器（家庭にあるブロードバンドルータの類）および関連サービスを提供する会社の新卒求人例を次に示します。

市場から求められている製品を設計、開発する初級レベルの JavaScript プログラマを求めています。プログラミング以外にも、自分の担当する商品仕様の明確化、適切な品質を維持するためのテスト（品質管理チームと連携してもらいます）、プロジェクト管理に必要な工程見積もりの提供といった作業も業務に含まれます。

応募者には Angular、React、Node を含む JavaScript プログラミング、Linux、TCP/IP をはじめとするネットワークプロトコル、REST などの通信フレームワーク、XML や JSON などのデータ表現の技能が求められます。プログラミングについては特にソケットの経験が必須です。学歴にはコンピュータ科学、コンピュータ工学、ソフトウェア工学、電子工学などの学士が必要です。

本求人は主としてソフトウェア開発を今後のキャリアと考えている新卒を対象としています。インターン経験があるとプラスです。

仕様の把握（設計）や品質管理（テスト）も任される点で、前記のインターンより責任範囲が広くなっていることがわかります。ただし、これらの作業には要求条件が加えられていないところから、初級あるいは見習いレベルであることも読み取れます。

Angular、React、Node は主にウェブ開発で使われる JavaScript の開発フレームワークで、これら全部とはいいませんが、ウェブ開発者に必須な言語系です。インターンの Microsoft 環境と同じように、言語が関連しているものでまとまってい

る点に注意してください。ウェブ開発者にはまた、上述のように通信技術の知識も必要です。数文字入力しただけで可能な選択肢をリアルタイムに変化させる予測入力のように、ウェブブラウザはサーバと常時通信しているからです。このように、ウェブ開発者であっても、プログラムそのものだけでなく、周辺の技術も把握していなければ十全に職務をまっとうできないことがわかります。

大学等でプログラミング、データ表現、ネットワークを学び、それなりの知識が習得できていればこれくらいできるような気がするかもしれません。しかし、これらを融合してひとつのプログラムを開発するには、それなりの経験が必要です。

● ソフトウェア開発者・中級（Software Developer, Level 2）

システム全体を視野に入れたソフトウェアの設計開発を担当します。要求分析などの上位工程やプロジェクト管理も支援し、開発チームの相談役も果たすなど、プログラミング以外の管理業務も少なからず含まれてきます。技術的には、周辺のサードパーティライブラリも含めてプログラミング言語は熟達の域に達しています。しかし、まだ全体の設計までは担当しません。業務経験は 5 年以上です。

難病に特化した小児医療を提供する病院兼研究機関（私立）の求人例を次に示します。

当機関のサイエンティストと共同し、クラウド上でゲノム解析を行うツール群を開発する中級レベルのソフトウェア開発者を求めています。業務には、ペタバイトクラスのデータを処理するワークフローの実装、クラウド上での各種ツール群の展開、ソフトウェア開発プロジェクトの主導です。

応募者には 5 年以上のソフトウェア開発、特に大規模な生命情報データの経験が求められます。学歴には工学、コンピュータ科学、物理化学などの学士以上が必要です。必須なプログラミング言語は Bash および Python で、これ以外にも Java、Perl、C/C++、Go、Rust のいずれか最低ひとつは必要です。ワークフローに関連しては WDL、CWL、NextFlow のいずれか、コンテナ技術は Docker です。システムについては Linux の技能が求められます。

中級レベルとなると、求められる知識が広くなると同時に、応用分野の経験が重

視されることがこの求人からわかります。

インターンと初級で求められていた言語が C/C++ や JavaScript だけだったのに対し、ここでは 7 点、最低でも 3 点が求められています。なお、Bash、Python、Perl はスクリプト言語で比較的簡単な処理を手軽に書くのに適しており、C/C++ や Rust はシステム開発に向いた言語です。単純なものから複雑なものまで、単体からシステム的なものまで広くカバーできる技能が求められているところに注目してください。

応用分野もゲノム解析経験が名指しで指定されるように、明示的になっています。生物の遺伝子情報を解明する生物学の一分野ですが、遺伝子も情報的にはいくつかの構成要素からなる文字列（巨大ですが）として扱えるので、コンピュータで統計的あるいは AI 的に分析できます。つまり、ソフトウェアエンジニアの出番です。しかし、ビッグデータを利用したゲノム解析をプログラミング開発の視点から把握するのは容易でなく、年季が求められることが読み取れます。

なお、この分野は生命情報学（bioinformatics）と呼ばれます。生物あるいは化学とコンピュータ科学の融合分野のサイエンティストの求人例は 4.3 節にあるので、そちらも参考にしてください。

ワークフローは、情報をいくつかのステップで順に処理していく流れ作業のことです。この作業も昨今ではシステム化されており、どのような順で何をどう処理するかはここに出てくる WDL（Workflow Description Language）、CWL（Common Workflow Language）、NextFlow といったワークフロー記述言語で規定します。ワークフローの知識が求められているということは、システム全体を効率よく動かすシステムエンジニア（第 2 章）の技能も必要であることを示しており、純然たる中級ソフトウェア開発者よりクロスオーバーな立ち位置です。ワークフローについては 2.5 節のワークフローアナリストのような専門職もあるので、そちらも参照してください。

● ソフトウェア開発者・上級（Software Developer, Level 3）

初級がシステムの一部、中級が単体のシステムを開発するのに対し、上級のソフトウェア開発者は複数のシステムからなるシステムのシステムを設計開発します。

単体のシステムはそれぞれ特定の目的を解決するために設計されます。たとえば、料金算定システムは顧客の利用情報や契約（プラン）のデータを収集して料金

を計算するという目的のために構築されます。顧客管理、営業管理、サポート支援、開発支援などのシステムにもそれぞれ目的があります。しかし、どれも単体では動作しません。料金算定システムが顧客管理システムからユーザデータを抽出して動作するように、他と連携することによってその機能を果たすのが通例です。AからBへのデータ転送がBの能力不足のために予定の時間内に完了しないといったようにうまく連携できなければ、システムは適切には動作しません。上級ソフトウェア開発者の役割は、システムのシステムが整合性、一貫性をもって動作するような大きな枠組みを設計するところにあります。

ソフトウェアは一度作成（リリース）したら終わりではありません。リリース後に発見されたバグの修正や新機能を追加した新バージョンがかならず出てきます。OS、プログラミング言語やそのライブラリ、データベースなど、そのソフトウェアが寄って立つ基盤でバージョンアップがあれば、それに対応しなければなりません。上級ソフトウェア開発者には、新バージョンをいつ出すか、そのバージョンにはどの機能を加えるか、入れられなかった機能はいつ入れるのかを決定する役割もあります。つまり、単一のソフトウェア開発工程だけではなく、それが何度も繰り