



情報処理技術者教育における デザイン教育の重要性と評価方法の一考察

有 田 宙 史

概要 情報処理技術者教育ではアプリケーション利用能力の開発やプログラミング能力の開発を積極的に行っているが、でき上がるものについての視点に欠けている。それ故に、コンピュータを用いた制作物から閲覧者が情報を適切に得られないばかりか、判読するのにも一苦労するものができ上がることもしばしばである。本稿では、この問題を解決するために必要なデザイン教育の重要性を論じ、また、その評価方法について考察する。

Abstract In information processing engineer education, we are actively developing application usage skills and programming skills, but they lack the viewpoint of what is finished. Therefore, it is not only the viewers can't properly obtain information from the work using a computer, but often they can make a hard work to read. In this paper, I discuss the importance of design education necessary to solve this problem, and also consider the evaluation method.

キーワード 情報処理技術者教育, デザイン教育, 商業デザイン, ユニバーサルデザイン,
デザイン評価

原稿受理日 2019年5月30日

1. は じ め に

現在は文系理系問わず情報処理技術者教育が当たり前に行われる時代である。ただし文系カリキュラムでは基本的に、コンピュータの利用、とりわけ広く社会で用いられる特定のアプリケーション利用法の習熟に重きを置いており、筆者の勤める近畿大学でも MOS (Microsoft Office Specialist) 試験対策を積極的に行っている。

また、単なるアプリケーションの利用に留まらず、よりレベルの高いコンピュータ利用を目的としてプログラミング教育も行っている。

だが、近年は COBOL や FORTRAN の時代と違い、プログラミングと一口に言っても、GUI の一般化で画面設計の自由度が飛躍的に向上し、ソフトウェアに求められる入出力の要件だけでなく、ユーザインターフェース (UI) デザインまで考慮しなくてはならなくなった。そのため、講義中課題で学生が制作するものに要件はクリアしていてもアプリケーションとして操作しにくかったり、画面が見にくかったりと問題が顕在化してきた。

本稿では、筆者が近畿大学、またそれ以前での専門学校講師経験を通じて感じたデザインに無頓着な学生のコンピュータによる制作物にありがちな問題点や現本業である IT エンジニアとして直面した事柄を基に情報処理技術者教育におけるデザイン教育の重要性と成績評価方法について論じる。

2. 情報処理技術者教育におけるデザイン教育の現状

文系カリキュラムの情報処理技術者教育におけるプログラミング教育では基本的にプログラミングに必要な基礎知識や文法を習得させる目的で行うが、習得自体が困難であることも事実である。これは、筆者が近畿大学で2013年より行っている講義での学生の課題達成率を見ても分かる。2018年度までで延べ226名中、達成できたのは85名で40%にも満たない。これは、課題出題から7～8週間を掛けて達成した総数で、1週目で達成できたものに絞ると7名で約3%、3週目までだと42名で約19%と極端に達成率は減る。専門学校で既に一通りのプログラミング教育を受けた後の生徒が、再勉強したいとの話があり個別指導でディスカッションしながらプログラミングを教えていた時に言われたことだが、困難さがもたらされるのはプログラミングを習得する上で必要な要素一つ一つは理解できても、その要素をどの局面で利用すればいいかを思いつかないからである。この事実は前述

の課題達成率とその推移からも証明されている。逆に言えば、それを思いつく学生はプログラミング課題を（時間はかかれども）クリアすることができる。ただし、特にスマートフォン等のモバイルデバイスアプリケーション開発においては、タッチデバイスならではのデザインを考慮しなければならないため、学生が比較的短時間にプログラミング課題が要求する仕様をクリアできても、アプリケーションとしては UI が足を引っ張っていることがある。

また、デザインはソフトウェア開発においてのみならず、経営学部を卒業する多くの学生にとって社会に出てから必要になる営業資料・プレゼンテーション資料等の作成でも必要とされるものである。しかしながら、デザインは一般的に芸術系の技能と分類・認識されるため、情報処理技術者教育においてはなおざりになりがちでアプリケーションの利用やプログラミング技術の習得は可能になっても制作物における表現方法の確立ができないため、他者にとって情報の取得が困難なものが作られる場合がある。プログラミングだけでなく、特に Microsoft Excel のような表計算ソフトでは項目見出し等のため簡単にセルの背景に色を付けられるが、背景色に文字色と競合する色を指定してテキストの判読が困難なものを一学生だけに留まらず社会人においても一作られることがしばしばある。

実際、現状で情報処理技術者にデザインが重要視されていないのは、IPA（独立行政法人情報処理推進機構）の実施する「基本情報技術者試験」を見ても分かる。当該試験は出題内容が2019年秋実施のものから見直しされるので見直し前（表1）と見直し後（表2）を比較してみる。

今回の見直しで主なトピックは、問7～問11で出題されるプログラミング言語が現代に合わせて AI 人材育成を目的に COBOL から Python に変更されていることである。全体としては問題数が削減されてはいるものの、問2～問4、問5で項目が増え、要求される知識の範囲が広がっているにも関わらず、このように本稿の執筆時の最新の見直しでも、デザインは UI 設計としても項目として挙がっておらず重要視されていないことが分かる。

しかしながら、問7～問11のソフトウェア開発分野に入っている表計算ソフトでも、先述の通り背景色の選定を誤るなどデザインの基礎から外れたものを制作するとマクロや関数の動作は問題なくとも、ユーザにとって判読が困難となったり、一見して、どこをどのように操作すれば良いか分からない、または UI 部品にアクセスしにくい（例えば大幅に画面をスクロールしないと操作ボタンに辿り着けない）などの問題を抱えた制作物になってしまう。情報システムでありながら情報を正しく伝えられないという本末転倒な事態を

表1 基本情報技術者試験見直し前

問番号	分野 (T:テクノロジー、S:ストラテジ、M:マネジメント)		選択方法		配点
			出題数	解答数	
問 1	T	情報セキュリティ	1 問必須		12 点
問 2～問 4	T	ハードウェア、ソフトウェア、データベース、ネットワーク	3 問	4 問	各 12 点
問 5	T	ソフトウェア設計	1 問		
問 6	M	プロジェクトマネジメント、サービスマネジメント	1 問		
問 7	S	システム戦略、経営戦略・企業と法務	1 問		
問 8	T	データ構造及びアルゴリズム	1 問必須		20 点
問 9～問 13	T	ソフトウェア開発 (C、COBOL、Java、アセンブラ言語、表計算ソフト)	5 問	1 問	20 点
合計			13 問	7 問	100 点

表2 基本情報技術者試験見直し後

問番号	分野 (T:テクノロジー、S:ストラテジ、M:マネジメント)		選択方法		配点
			出題数	解答数	
問 1	T	情報セキュリティ	1 問必須		20 点
問 2～問 4	T	ソフトウェア・ハードウェア、データベース、ネットワーク、ソフトウェア設計	3 問	2 問	各 15 点
問 5	M・S	プロジェクトマネジメント、サービスマネジメント、システム戦略、経営戦略・企業と法務	1 問		
問 6	T	データ構造及びアルゴリズム	1 問必須		25 点
問 7～問 11	T	ソフトウェア開発 (C、Java、Python、アセンブラ言語、表計算ソフト)	5 問	1 問	25 点
合計			11 問	5 問	100 点

出典：IPA プレス発表 基本情報技術者試験における出題を見直し（2019.1.24）

引き起こすこととなる。

それでは、デザインに無頓着なまま制作したものとは一体どのような問題を抱える可能性があるのだろうか。次章で実際の事例を示していく。

3. 学生の制作課題で起こったデザイン問題事例

3-1 ユーザインターフェースデザインにおける問題

デザインに無頓着に制作したものとの問題事例として、筆者が近畿大学経営学部で担当する「iPhone アプリ開発入門」で出題している課題の完成画面（図1）と実際に学生が制作したものの再現（図2）をご覧頂きたい。課題の仕様詳細は割愛するが、このアプリケーションは画面下部のスライダーの○を動かすことで、画面上部のラベル部分（背景に帯のある箇所）にスライダー内の○の位置に従った結果を表示するのが数ある仕様の内の一つの要件である。課題の提出は講義の時間内での目視の完成チェックとし、プロジェク



図1 課題完成画面



図2 学生の制作課題の再現

トフィールドを受領する形は取っていないため、図2は実際の提出物を再現したものである。

図1と比べて、図2では、スライダーが画面幅一杯に配置されているのが分かる。図は共に Apple 社が提供する macOS/iOS/tvOS 用の開発環境「Xcode10」に付属する iOS Simulator のウインドウであるが、このバージョンではウインドウが iPhone を模しており、実機で表示したときの状態を確認すると、それに伴い本稿での論考に都合がよいためアプリケーションの画面だけでなくウインドウごと引用している。

図は macOS 上で動作するシミュレータではあるが、マウスカーソルでのオペレーションでもウインドウ枠が邪魔になってスライダーの左右両端付近で正確に○をドラッグすることが難しい。図でも分かるように実機においては両端の○にアクセスしようとするとベゼルと液晶を指先が跨ぐため、操作のしにくさは言わずもがなである。近年はベゼルレスのスマートフォンも増えており、iPhone も iPhone X でベゼルレスを実現しているため、指先が半分近く本体からはみ出た状態から操作を要求され、さらにアクセスおよび操作が困難となる。この困難さを招く UI であることは設計上問題がある。

3-2 ページデザインにおける問題

次に、アプリケーション UI ではなく、ページデザインで起こる事例を見ていきたい。2013年秋に、ある学生グループがデジタル・サイネージ（電子看板）の企画・デザインを行い、それに指導を要求されたことがあり、問題点を明らかにし、それに基づいて修正点を解説しながら Adobe Photoshop を用いて20分ほどで同要件のものを簡単に作成して見せたことがある。

まず、学生グループが作成した電子看板のデザイン案（図3）を見ながら、この制作物が持つ問題を明らかにする。



図3 学生が制作したデザイン案

この制作物ではデザイン上、細かいものを除き、以下の致命的な問題点が見受けられる。

- ①何の情報なのか分かりにくい
- ②背景がうるさくて文字が読めない

これら問題点を順になぜそうってしまったかの原因を追っていく。

まず①について、全てのフォントサイズやウエイトが全く同じで、強調箇所がないために、画像から推してハンバーガー店のものであること以外、これがいったい何の情報であるかが一見して分からない。これは筆者が専門学校時代からデザイン講義をして学生とディスカッションを繰り返した結果、経験的に判明していることであるが、制作した本人

達が日常的に目にする印刷物やウェブページ等で、フォントサイズやウエイトがどう使用されているかに全く無頓着で適切なフォントの用法が思いつかないために、このような状態になる。右下の QR コードが小さすぎるのも、強調すべきところの理解がないからで、このサイズでは電子看板の閲覧者が QR コードを活用するのが難しい。このプロジェクトの要件として画面サイズはフル HD（1920×1080px）であるが、20インチ程度の画面であったとして QR コードの大きさは1cm 四方程度のものになってしまう。QR コードリーダーで読み取るにもモバイルデバイスをかなり近づかねばならず不便である。

②については①に引きずられて起こる問題である。背景がうるさくフォントサイズやウエイトが適切に設定されていないので文字が読めない。本人達もなぜ納得のいくものにならないのかを悩んでいた。こうしたデザイン上の問題が起こる理由については、先述の通り、デザイン講義で散々目の当たりにしてきたので、それをそのまま学生に「背景が白くて余白が多すぎるから寂しいなと思って適当になんとなく背景に画像を貼り付けたでしょう？」と問うた。これはまさに正鵠を射ていた。

つまり、作ってみたはいいものの適切なフォントの用法に思い至らず、さらに QR コードも小さいため余白が大きくなり、間が持たないので、という理由でなんとなく背景画像が用いられたことになる。これは学生に限らずデザインに無頓着な人々が陥る顕著な事例である。非デザイナーである社会人が個人で作ったウェブサイトに意見を求められたこともしばしばであるが、やはり同様の問題を抱えていた。それも派手な背景を用いれば綺麗になったと錯覚までしていることがある。そして、この“なんとなく”で配置（レイアウト）するのはデザイン論的に間違いである。なぜ間違いであるかは後述する。

3-3 ページデザインにおける問題の解決

先に見てきたページデザインにおける問題点を踏まえて解決すべく、筆者が出した解答例が図4である。ただし、先述の通り、学生の限られた休み時間をできるだけ潰さず例として見せるためだけに、デザインの考え方を逐一解説しながら短時間で簡易に作成したもので、元のデザイン案（png 画像）からの画像の切り取りが杜撰であったり細かい位置あわせを行っていない上に、ここからさらに突き詰めて完成させる必要はある。

解答例では、図3で背景に使われた画像に従って、ベースカラーを青系で統一した。これは同じような色を使っても全くイメージの違うものになることを解答例として端的に示すためで電子看板として色が適切かは二の次としている。何の情報かを一目瞭然にするため店舗名のフォントサイズを大きく、背景にフォントカラーと彩度を取った背景色を用い



図4 筆者の提示した解答例

た単色の帯を配置して強調し、その他、トピックタイトルはフォントウェイトを重くしてある。また、背景の画像は取り除き、閲覧者の視点を適切に引くようにトピックに枠を設けたり、写真等を配置している。QRコードもモバイルデバイスのリーダーで撮影しやすいように大きくしている。フォントサイズは前述の通りフルHDの液晶画面を想定して選定している。ただし、少し遠目でも見えるように全体的にもっとフォントサイズを大きく取るべきかも知れない。これは電子看板の利用者が、どの距離からどの程度の画面サイズのものを読覧するかを企画として想定し、それに基づき決定すべきことである。さらに重複していた不要なムービー操作UI（「再生」「停止」「初めから再生」ボタン）も削除して閲覧者がどちらを操作すればいいか迷わないようシンプルにしてある。そもそも電子看板である以上、ムービーの操作ができる必要があるかどうかは疑問である。この他、一定していない外周の余白の取り方も修正してある。

こうして、各種情報に必要な意味付けをフォントサイズ等で行うだけで、学生がしたように背景に画像を配置しなくても間延びなどすることはなく、かつ間延びを防ぐために小細工をしているわけではないことが要点である。

4. 情報処理技術者教育で必要なデザイン教育

ここまで、デザインに理解のないまま制作したものの問題と、その解決法を見てきたが、

なぜこうした問題が起こるのだろうか。これらの原因は、『3. 学生の制作課題で起こったデザイン問題事例—3-2 ページデザインにおける問題』で述べたように、日常的に目にするデザイン物の構成が頭に入っていない、あるいは普段からデザインを考慮したことがない、もしくは根本的にセンスが無いために起こる。つまり、デザイン手法を体系化して教育すれば、例えセンスが無くとも、これらは一定のレベルで解決でき、ただアプリケーションが利用できる、もしくはプログラミングができる、というレベルに留まらない人材が育成可能である。

それでは、デザイン手法に触れたことのない人々にどのような教育をすればよいだろうか。この場合に求められるのは商業デザインの一部であり、肝要なことは限られている。ここで述べる商業デザインとは、キャラクターデザインのようにイラストを描くような特殊技能を必要とするものではなく、タイポグラフィやレイアウトを論理的に用いることで可能なデザインのことである。それ故、単に情報を見やすくする、UI にアクセスしやすくする、という目的であれば、やらなければならないことは数値化・体系化できる。

閲覧者や利用者が画面から情報を得やすいものにするには、先に電子看板の事例で示した通り、情報の優先度に従ってフォントサイズの選定、色の選定、余白の取り方さえ押さえればよく、これらは一定の基準に基づいて判断が可能であり、かつ商業デザインのセオリーの一部である。セオリーがある以上、“なんとなく”でレイアウトすることは『3. 学生の制作課題で起こったデザイン問題事例—3-2 ページデザインにおける問題』の最後に述べたように間違いである。画面のベースカラーを決定する色の選定を除けば芸術的センスを求めずともロジックで誰でも習得可能となる。これを踏まえ、以下のような項目を教育することで、これまで述べたような問題は大部分で回避できる。

①各所に使用する適切なフォント・サイズ・ウエイト

一番目立たせるべき画面内のタイトルに使用するフォントに適切であるのはゴシック系フォントであり、サイズも本文より最低1.3～1.5倍ほど取り、ウエイトもボールドとする。より目立たせたい場合は、下線を付けたり、背景に色付き帯を配置する。見出しに使うフォントはウエイトをボールドとし、本文と同じか本文以上のサイズにするならタイトルに使用したサイズよりは小さいサイズにする。本文は画面サイズに依るが、サイズを10～12pt/12～14px程度とし、ウエイトはノーマルを使用する。また、本文の行間をフォントサイズの0.2～0.3倍ほど空ける。

②書体の特性を知る

見やすいものを作るには書体の特性も知っておく必要がある。ゴシック体は見出しに使いやすいがウェブ等の PC モニタに出力する場合は本文にも利用することが望ましい。明朝体は横向きの線が細いので紙に比べて解像度に乏しい PC モニタでは見づらい場合がある。印刷物であっても紙の質や出力機によっては横線がかすれるため明朝体の使用には注意が必要である。

③レイアウトのための余白（マージン）の取り方

文字や画像を画面/紙面の端ギリギリに配置せず、画面サイズに対して必ず上下左右に5%程度の余白を設ける。画面/紙面を横2カラム以上のレイアウトにする場合、それぞれのカラムの間に余白を設ける。また、テキストの表示枠を設けるなら、枠からテキストまでの余白を設ける。それぞれの内容に適切に余白がなければ非常に見づらいものとなる。

④背景色の彩度に対する適切な文字の彩度

文字と彩度が近い色を背景に配色してはならない。文字が黒系なら白に近い明るい背景にする。文字が白なら背景を黒系の暗い色にする。背景が写真等で、やむを得ず彩度が近い部分が出て読みにくくなる場合は、文字に影を付ける、文字の縁取りをする、背景の写真とテキストレイヤーの間に半透明のベタ塗り枠を設けてコントラストを確保する等、背景と同化しないようにする。

⑤ベースカラーの選定法

色に対して人が抱くイメージを理解する必要がある。白は清潔感を、黒は高級感を、赤は躍動感を、青はクールさを、緑は優しさを、と、こうしたイメージから配色を選ぶとよい。1つの色が持つイメージは、ここに挙げた例だけでなく他にいくつもあり、その中から選択する。企業のデザインなら企業イメージや、すでに企業ロゴがあるなら、ロゴデザインから色を決定するのも一つの手である。

⑥ MS 明朝/MS ゴシックファミリーは極力使用しない

これらフォントファミリーは古い時代の解像度の低い CRT で見やすいようにという思想だけで設計されており、タイポグラフィにはそぐわない。ひらがなも文字の太さや線の湾曲デザインの統一がされていないため読みにくい(図5)。僅かな違いではあるが、こうした統一感の無さによって読み手がはっきりと意識せずとも潜在的に読みづらさを与える。Windows8.1以降ではウェブページなら「メイリオ」、それ以外ならゴシック系は「游ゴシック」、明朝系は「游明朝」を優先的に使用し、Mac では

ゴシック系は「ヒラギノ角ゴ Pro」，明朝系は「ヒラギノ明朝 Pro」を使用する。ただし，Windows の場合，フォントレンダリングが非常に稚拙なため，あくまで PC モニタ上で見る上では10ポイント前後の小さい文字に限り，ビットマップフォントになっている MS 明朝/ MS ゴシックファミリーを使用した方が見やすい場合もある。

<p>えおや いうの</p> <p>「い」「う」「の」は他に比べて 僅かに文字が太い。</p>	<p>こたに</p> <p>「こ」だけ他より直線的で 似た形状を持つにも 関わらず統一感がない。</p>	<p>きさ</p> <p>左の例と同様に 「き」は「さ」より直線的。</p>
---	--	--

図5 MS ゴシックの問題点

⑦ブロックの幅を合わせる

ボタンやバナーが縦に並ぶ場合には横幅を統一し両端を揃えておく。それらのキャプションも字間を詰めたり空けたりして両端を揃える。また，テキストブロックでは，段落を均等配置にしてブロック右側の文字の端が揃うようにしておく。右端がずれると読みにくくなる。字間も1行に収まりの良ように詰めたり空けたりする。

⑧なぜそうするのかを常に意識しながら制作する

なんとなく画像を配置する，なんとなく文字サイズを決める，なんとなくおしゃれな感じがするからフォントを決定する等，“なんとなく”で決定してはならない。フォントの決定は読みやすいものであることを最優先として，英語フォントならデザイン性の高い筆記体のようなフォントを本文に使用することは極力避ける。デザインは人に情報を伝えやすくするために必要な技術で，まず目を引かせたい箇所，それからじっくり読んでもらいたい箇所等のレベルを自ら意識して意味付けしフォントや画像，色を決定しなければならない。

勿論、マージンやベースライン等の専門用語教育も必要だが、このようにデザインでも一定の数値化が可能なもの、また体系化できるものを選択すれば、芸術的センスという曖昧なものに頼ることのない教育が可能である。一定の数値化が可能な商業デザインである故にデザインに無頓着な人々でも、ロジックで制作したものが、ある程度、他者に訴求できることを理解すれば、自ら違和感を覚えながらも“なんとなく”で手を入れて、より見づらくなるような泥沼に陥ることもない。

5. これからのソフトウェア開発におけるデザイン教育

さて、ここまで述べたところで、昨今では利用者の公平性への観点からユニバーサルデザインが重視されるようになってきている。実のところ既に述べた事柄は、ユニバーサルデザインについて考慮しておらず、あくまで情報が埋没しないためのデザインの基礎的なことを論じているに過ぎない。しかしながら、コンピュータでの制作物においては、ユニバーサルデザインの理念である「誰もが使いやすい」を実現する重要性は高い。我が国において、ユニバーサルデザインはバリアフリーと混同されがちであるが、本来、「誰も」の対象に健常者と障害者の区別はなくバリアフリーとは趣を異にする。ユニバーサルデザインは1985年に米ノースカロライナ州立大学のロナルド・メイスによって公式に提唱され、今日では以下の7つの原則が定義されている（表3）。7原則には合計30個にも渡る指針があり、実際に教育する際には必須事項として重要であるが冗長になるため割愛する。

表3 ユニバーサルデザインの原則（抜粋）

原則1：公平な利用	どのようなグループに属する利用者にとっても有益であり、購入可能であるようにデザインする。
原則2：利用における柔軟性	幅広い人たちの好みや能力に有効であるようデザインする。
原則3：単純で直感的な利用	理解が容易であり、利用者の経験や、知識、言語力、集中の程度などに依存しないようデザインする。
原則4：わかりやすい情報	周囲の状況あるいは利用者の感覚能力に関係なく利用者に必要な情報が効果的に伝わるようデザインする。
原則5：間違いに対する寛大さ	危険な状態や予期あるいは意図しない操作による不都合な結果は、最小限におさえるようデザインする。
原則6：身体的負担は少なく	能率的で快適であり、そして疲れないようにデザインする。
原則7：接近や利用に際する大きさと広さ	利用者の体の大きさや、姿勢、移動能力にかかわらず、近寄ったり、手が届いたり、手作業したりすることが出来る適切な大きさと広さを提供する。

出典：ロナルド・メイス（1985）

これらのうち、情報処理技術者教育において特に重要なのは原則 3～5 であると考え。先述した筆者の担当する講義である「iPhone アプリ開発入門」で出題している課題で仕様はクリアして動作に問題はないものの、完成画面が提示されているにも関わらず、ボタンキャプションを「a」等と全く意味をなさないものにして提出した学生がいる。これは先例の UI の配置に問題を抱えているよりも、もっと深刻な問題で原則 3～5 を大きく逸脱している。機能さえ実現できていればよい、という合理性が突出しており他者に見せる感覚がない。こうした感覚の場合、ユニバーサルデザインの観点には遠く及ぶことができない。既に述べている通り、通常の商業デザインの多くは感覚ではなくロジックや合理性をもって行うことができるが、ユニバーサルデザインでは、7 原則で示されているように時には合理性を廃し、「誰もが使いやすい」ことを優先する必要がある。原則 3～5 を満たすには、ボタンと分かるデザイン、どのような機能を持つボタンであるかが一目瞭然のキャプション、誤タッチ・誤クリックの可能性を下げるための UI 部品間の適切な余白、ダイアログ等で「OK」と「キャンセル」ボタンの位置が都度入れ替わったりせず、ウインドウ枠からの距離も一定に保つこと等、様々な事柄を考慮する必要がある。また、原則 6 にある身体的負担においては、強い光が苦手な人のために背景を暗く、文字を明るくする Windows10 や macOS Mojave 等のモダン OS で導入されているダークモードのようなデザイン切替機能の導入も必要である。実際に筆者が開発した macOS 用アプリケーションで、テキストを表示するウインドウ配色の変更機能追加を要求されたことがあり、これにはウインドウ背景が白で文字が黒い場合、目が痛くなるためという理由があった。

強い光が苦手な人だけでなく、視力が弱い人のためにウエイトを重くしたユニバーサルデザインフォントを利用することや過度に小さな文字を使用せず、可能であれば文字の大小を選択できる UI を付与する。また、我が国において男性の 5 %（20人に一人）、女性の 0.2%（500人に一人）見られるという先天色覚異常者に配慮して配色をすることも必要になる。こうした人々は青系と黄色系の組み合わせを混同することはないので、強調したい場合は、これらの色を組み合わせることや明度や彩度に差を付けたり、黒や白の無彩色で縁取り等を施すことで何が表示されているのか、あるいは情報の重要度が把握しやすくなる。

6. デザイン教育における成績評価とルーブリック表

これまで見てきたように、フォント選定と利用法、適切な余白を用いたレイアウト、テ

キストと背景が競合しない色の選び方を押さえるだけで、コンテンツが同じでも全く違うものになる。UI であれ、ページデザインであれ、ユニバーサルデザインであれ、学生に問題点の把握と、解決法の取り方を促すために、ここまで述べたことのまとめと一考察としてループリック表の一例（表 4）を設定してみた。

表 4 デザイン成績評価のループリック表（例）

	4	3	2	1
フォントを選定する力	情報の優先度に従ったフォントサイズ・書体・ウエイトを使用している。	情報の優先度に従ったフォントサイズ・書体は選んでいるが、すべて同じウエイトのフォントを使用している。	情報の優先度に従ったフォントサイズは選んでいるが、すべて同じ書体・同じウエイトのフォントを使用している。	すべて同じサイズ・同じ書体・同じウエイトのフォントを使用している。
レイアウトをする力	余白が一定基準で設けられ、文章の行間がフォントサイズの0.2～0.3倍ほど空けられている。	余白はあるがバラバラで文章の行間がフォントサイズの0.2～0.3倍ほど空けられている。	余白はあるがバラバラで文章の行間が広すぎ/狭すぎる。	カラム間の余白がなく文章の行間が詰まっている。
色を選定する力	フォントの色相・彩度と適切に合わせた背景色を使い、部分的に見づらい箇所は適切に縁取り・影を設け、テーマのイメージに沿った色を使用している。	フォントの色相・彩度と適切に合わせた背景色を使っているが部分的に見づらく、テーマのイメージに沿った色を使用している。	フォントの色相・彩度と同じ背景色を使い、テーマのイメージに沿った色を使用している。	フォントの色相・彩度と同じ背景色を使い、テーマのイメージにそぐわない色を使用している。

『2.情報処理技術者教育におけるデザイン教育の現状』で確認したように、基本情報技術者試験では、デザインへの理解を求められていないため、このループリック表は少なくともこの分野において過去にも活用されていない。

なによりも、学生には商業デザインは感性で行うものではなく、ロジックで行うものであることを理解させるのが重要である。ロジックのできるのであれば、（芸術的センスがない）自身にも可能ではないかと自信を持たせることができる。与える課題については規模にも依るが、大きなものであればグループワークでの制作でも構わない。考えられる課題としては、プログラミングであれば制作するアプリケーションの UI 設計を行わせる、プレゼンテーションであれば、学生に特定の企業を想定させて、事業ドメイン紹介のスライドを誰にでも分かるように制作させる等、がある。これら評価においては、グループワークであれ、パーソナルワークであれ、『4.情報処理技術者教育で必要なデザイン教育』

中で述べた『⑧なぜそうするのかを常に意識しながら制作する』の通り、最終的に「なぜそうしたか？」「どう考えたか？」が重要であるのでディスカッションも欠かせない。

ただ、ユニバーサルデザインに関しては健常者が見たときにも見やすい、あるいは退屈しない（最後まで見る気になる）配色のバランスを取るセンスも必要となるため、評価に関してはデザインの基礎ができていることとユニバーサルデザインを考慮した上での美しさとは別にしなければ体系的に評価できない。大きな文字はバタ臭く、小さい文字はスマートに見えるため美しさだけ優先するとフォントサイズは小さいものを選びがちになるが、ユニバーサルデザインでは全体的な美しさより、フォントの見やすさ、コントラストの使い方等、推奨しうる指針が既にあるため、それに基づいていれば正解とする。時間が許すのであれば、同じ課題で通常のデザイン版とユニバーサルデザイン版の両方を制作させて、それぞれ評価すれば学生にもデザインの意味をより深く理解させることが可能である。勿論、ユニバーサルデザインであるかどうかには関わらず、色の問題は定量化が難しいため、理由がはっきりしていて、『4.情報処理技術者教育に必要なデザイン教育—⑤ベースカラーの選定法』で述べたような色の持つイメージを把握した上での選定基準に矛盾がなければ正解とする。

7. お わ り に

制作物は他者に見せることが主題にあり、多数の人が見て、制作者が伝えたい情報の流れ・優先度、そこに掲載された情報の意味が伝わるように作らなければならない。こうしたことを実現するためにはデザインの基礎を等しく学ぶ必要があるが、コンピュータを用いる場合、何らかの制作を行うにも関わらず、情報処理技術者教育ではあまり重視されないデザイン教育の重要性と評価方法について論じた。本論中で述べてきたように、情報処理技術者教育におけるデザイン教育の重要性は、見た目や UI 設計でユーザに不便を強いることのないように制作物の総合的な完成度を高められる人材を育成できることにある。デザインは専門のデザイナーに任せて開発者は開発だけすればいいではないか、という反論もあるが、仕上がったデザインをデザインの基礎が理解できていない者が見てソフトウェアに落とし込むと、特に動的な部分でデザイナーの意図を理解できずに正確なデザインの再現ができなかったり、適当に余白を設けたり、適当にフォントを使用してみたりと全体的に統一感のないものになってしまう。これは、筆者が20年以上に渡って IT 業界に携わってきた中で度々経験したことで、その度に戻り工程を経ることとなる。それ故に、

デザインの基礎を学んでおくことは情報処理技術者として有利であると考えられる。しかしながら、コンピュータプログラミングの技能や、Microsoft Office スイートの扱い方を学びたい学生にとって、デザインだけに特化した講義は訴求しにくいだろう。プログラミング論のようなプログラミングの普遍的な基礎を学ぶ講義ではともかく、スマートフォンアプリケーションの開発を目的とした講義では、UI 設計は切っても切り離せないため、してはいけない UI の配置法やボタンキャプション等の見た目では何をすればいいかわかることが必要であるとの教育をプログラミングと同時に行うのが望ましいと思われる。プレゼンテーションや Microsoft Office スイートの習得講義では、アプリケーションの操作法と共にスライドや文書、表の見やすさ、目の引きやすさの理解を促すため、全く同一サイズのフォントを用いた場合と重要度に合わせてサイズやウエイトを変えた場合等の実例を持って教育すればよいのではないだろうか。

最終的には UI やページデザインのロジックを学び、ただアプリケーションが利用できたりプログラミングができるだけではない、より高度な人材育成のための体系的なカリキュラムが実現できるかどうかが課題である。

参 考 文 献

- 独立行政法人 国立特殊教育総合研究所 (1997) ユニバーサル・デザインの原則
http://www.nise.go.jp/research/kogaku/hiro/uni_design/uni_design.html (accessed 20 May 2019)
- 公益財団法人 日本眼科学会 先天色覚異常
http://www.nichigan.or.jp/public/disease/hoka_senten.jsp (accessed 21 May 2019)
- 公益社団法人 日本眼科医会 3.見え方はどんな感じなの？
<https://www.gankaikai.or.jp/health/50/03.html> (accessed 21 May 2019)
- 公益社団法人 日本眼科医会 8.カラーユニバーサルデザイン
<https://www.gankaikai.or.jp/health/50/08.html> (accessed 21 May 2019)