

治郎吉商店インターンシップレポート

電気通信大学 情報工学科 3年
長谷川 伸吾
2008. 9. 12.

治郎吉で学んだこと

● 一般

○ 他人に訊く

わからないことがあったら、ひとりで長々と考えずに人に訊かなければならない。
「質問ができないようなヤツはバカにされる」とのこと。
プログラムを組んでいて思うようにいかなかったときや、不具合を発見したがどこを修正すればいいかわからないとき、そのほか自分では気づかないこと、理解できないことは、ひとに訊くとすんなり解決することがほとんどであった。
学校では自分で考えぬくという姿勢が当たり前になっていたのも、はじめは全然質問ができていなかった。
わからなかったら訊く。インターンで学んだ最重要事項である。

○ 途中経過の報告・方向性の確認

どんな仕事をするにも意識しておかなければいけないこと。早めの段階で自分がしていることの方向性を指導者に確認して、方向を正さなければならない。
仕事が一通り終わってから「これでいいですか？」ときいて、「これじゃダメ」と言われたときのショックは計り知れない。
一度仕事に集中してしまうと忘れがちになってしまうので気をつけよう。常に周りからの意見を求め、自分だけでは仕事をしないという姿勢が必要である。

○ コミュニケーション

質問や報告をするとき、文章を書くときに自分の言いたいことを伝えるのはとても難しい。
何度も「日本語がヘタ」「それじゃ伝わらない」と言われ、改めてコミュニケーション能力が低いことを感じた。
特にプログラムに関する内容は専門用語の使い方が難しいので、遠まわしでない言い方をしないように気をつけたり、実際にプログラムが動いているところを見せられるなら見せたほうが早いということを学習した。
これからの自分に一番必要だと思った能力である。

○ 仲間

同じ立場で働く、自分と同じインターン生。悩みあい励ましあうことで助けられた。ひとりだけでインターンをしていたら心身ともにボロボロになっていたことだろう。

● テクニック

○ ホームポジション

ホームポジションをキープしてのタイピングが出来ていなかった自分は、ホームポジションを重視する治郎吉の方々にMacパソコンで鍛えるように言われた。Macはホームポジションを鍛えるのにいいとのこと。
合言葉は「左手親指はコマンドキーに」。

○ ネットの資料は画面で読むな、印刷して読め

インターネットで得られた資料を読むならモニターで見ずにプリントアウトして読む。こうすることで調べた資料を指導者も見れるようになる。
学校で印刷枚数を超過して怒られた自分にとっては印刷し放題な状況が地味に嬉しかった。

● 学校と会社の違い

○ わからなかったら訊く

学校では自分で考え抜くという姿勢が当たり前と感じていたが、会社では5分考えてわからなかったら人に訊くという姿勢でなくてはならない。
「質問ができないようなヤツはバカにされる」とのこと。
会社では自分の能力を把握して行動しなければならない。「時は金なり」なのでムダに時間を浪費してはいけない

のだ。

インターンが始まったころは学校のやり方が身についていたし、「質問するのは恥ずかしい」、「『そんなこともわからないのか』と思われたくない」、「迷惑になるのではないか」、「なんとか自力でやりとげたい」、「訊くのは緊張する」、「怒られるのではないか」という思いがあって、質問することをためらっていた。

治郎吉で働くことに慣れたして、社員の性格も大体わかってきて、「わからなかったらすぐ訊いて」と何度も言われたことで、ようやく開き直ってどんどん質問ができるようになった。

教訓:「質問しないことを怒られるより質問して怒られる」

- プログラムは効率よりもわかりやすいものを

学校では、たとえばメモリを節約するために変数を使いまわしたり、処理速度の向上のために複雑な形でプログラムを書くことがしばしばある。治郎吉ではこれはNG。

学校の教員は、実はプログラミングを専門的にやっているわけではないので、実際にソフトウェア会社でやっているようなコーディングには疎いとのこと。しかもメモリ容量が小さかった昔の時代を踏襲しているため、先の例のようなことをやっているようだ。

会社ではひとりだけでプログラムを組むことはないの、他人に見せても伝わるようなコーディングをしなければならない。

教訓:「学校のやり方は会社では通用しない」

治郎吉のやり方

● ファイル保存方法

- 作成したファイルはファイルサーバーに保存

会社のものとして共有できるようにする。ファイルの受け渡しはここを通じて行う。

学校ではファイルの共有などせず、自分のものは自分のものとして大事にしまっておくという秘密主義な考えなので、はじめは違和感があった。

- こまめなバックアップ

バックアップをとるという習慣がなく、同じファイルは上書き保存するというクセがついていた自分には新鮮だった。

仕事ではいつデータがおかしくなるかわからないのでバックアップをとっておくことはとても重要。

インターン後には、自宅での作業でもこまめにバックアップをとるようになった。

- ファイル名には日付や番号を入れる

「こまめなバックアップ」に関連する。

最新のファイルが一目でわかるようにする。

インターン前にもそれなりにやっていたが、徹底してはいなかった。バックアップをとっていくといくつも同名ファイルが出来るので番号をつけて区別するようにする。

● プログラミング

- コーディング

治郎吉にはコーディング規約が定められており、変数名やコメントの書き方、モジュール化の方法などが細かく指定されている。

ひとりだけでプログラムを読み書きすることはありえないので、コードの書き方が統一されている必要がある。

いくら処理速度が速くて意図したとおりに動くプログラムでも、コードが読みにくかったりコメントが薄ければ作成者以外の者には何をしているプログラムか読み取る手間が大きいし、モジュール化されてくなくだらだらと長いコードであれば再利用性がない。このようなコードは産業廃棄物と呼ばれる。

目指すは他のプロジェクトでも利用できるような、会社の財産となるようなコード。

- フローチャート

プログラムを書く前にやること。モジュール化の手助けになる。

このフローチャートを書くことで「実際にプログラムを書く時間というのはすごく短い」のだということを聞き、1行ごとに考えながらプログラムを書いていた自分はショックを受けた。

自分は情報系にもかかわらずフローチャートを書いたことがほとんどなかったのになかなか大変だった。

プロのやり方を1つ学んだ。

- 1行書いたらコンパイル

プログラムは1行書くごとにコンパイルをしてエラーがないか確かめる。

まとめて書いてまとめてコンパイルしてエラーが多発して修正が大変になる。という自分には、このプロのやり方に大きなショックを受けた。

- テスト

プログラムに新しく文を記述するごとにプログラムを実行して、意図したとおりに動くかどうか、不具合がおこらないかどうかテストする。

不具合がおこったらただちにプログラムを修正をする。不具合がある状態で次の作業にはってはダメ。

治郎吉での仕事

● 日課

- 日報

その日に自分がやる予定のこと、やったこと、明日やること、感想を書き、メモとともに一日の仕事の終わりに指導者に提出する。

はじめは殴り書きのような汚い字で書いてしまったが、作業報告書となっていることに気づき途中から丁寧に書くように努めた。
- メモ

調べたこと、教えてもらったこと、気づいたことなど、仕事中に必要となったことを逐一メモする。あとで見返せるようにファイルに閉じる。

メモとはいえプロジェクトに関係するものは専用のファイルにはさみ、会社の財産となるので、読める程度の字で書く。
- 弁当の注文

日課。社員およびインターン生(自分含む)から注文をとり、お店に電話で発注する。

毎年インターンの時期に注文するお店は同じようだが、今年は石油高騰のせいか値上がり&盛りが少なくなったようだ…
- 掃除

お客様が来訪するときや、床にゴミが散っているときにする作業。職場は清潔に。

● 雑務

- 新聞の切り抜き

指導者が忙しいときの指示待ちや古い新聞が溜まってきたときにやる作業。

社員が蛍光ペンでマークした記事、自分が興味を持った記事を新聞から切り抜き、閲覧用とする。

社員間の情報共有、記事すべてを見るができないときの時間の節約、社会の情勢把握に役立ちそうだ。
- お茶だし

お客様がいらっしゃったときに麦茶をお出しする。

出すまでのスピードと出すタイミングが重要。
- コピー

「コピーとっという」のフレーズにより、コピーをとる作業。

インターンが始まった頃はこの作業でさえ緊張した。

● 使用パソコン

- ダイナブック(Dinabook)

治郎吉で使われているパソコンはほとんどMac。しかしWindows環境での仕事メインである自分は、インターン生に不人気No.1との声高いダイナブックを使用。

音がうるさいわすぐ熱くなるわで不満だったが、マイパソコンよりもキータッチがよかったので愛着がわき、最後まで使用し続けた。以降のインターン生にオススメはしない。
- マック(Mac)

治郎吉オリジナルソフトであるMac版文庫番の動作確認で使った程度。

コマンドキーを駆使するのが楽しく、使い慣れればとても便利ようだが、ダイナブック(OSはWindows)をメインで使っていた自分はその境地まで至れなかった。
治郎吉はMacがメインパソコンだと聞いており、せっかくだから今回のインターンでMacの使い方も覚えたいと思っていたので、少し残念だった。

● 仕事(Windows 版文庫番の機能実装)

○ 概要

すでにMac版として開発されている治郎吉オリジナルソフト「文庫番」のWindows版を作成する仕事。基礎となる部分を用意してくれており、それに機能を追加していく。

「文庫番」とは『ごく短いメモや電子メールからホームページの内容や文献データのような長文まで、簡単に蓄積し管理しながら、再利用しやすいテキスト・データベースを育てていくことができ』るソフトである(治郎吉ホームページより引用)。

「エディタをつくりたい」という面接での言葉により任された仕事である。

○ 本文保存

最初に実装した機能。リッチエディットに書かれたデータを保存する。

まだGUIアプリケーションのプログラミングがさっぱりわかっていない頃だったため、しどろもどろしながらの作業。Mac 版文庫番のソースコードや、文書名の保存のプログラムを参考にして、コピー & ペーストによる作業がほとんど。

内部関数がたくさんあって驚いていたが、その使用方法是コメントや関数の引数名を見てなんとなく理解できていた。今にして思えばあまり知識がない自分に対してもこのように理解の大きな手助けになっていたので、コーディングの重要性がわかるものだった。

しかし、結局RTFデータをバッファに送るやり方がわからず久永さんに助けをもらうことになった。RTFデータをバッファに保存するにはコールバック関数という聞きなれない関数を使う必要があった。早くも学校と会社とのプログラミングの差を痛感していた。

悔やまれることに、次の不具合をインターン期間中に解消できなかった。

『大きなデータを保存すると次に本文を読み込んだ際本文が空になっていたり本文の一部にゴミデータが混じっている』

○ 本文印刷

本文に書かれている内容を印刷する機能だが、ソースコードはまさしく産業廃棄物である。

機能としても印刷すると右端がきれてしまう不具合があったりページ範囲の指定ができないというように中途半端なものになっている。

途中経過をしばしば見てもらわず、自分だけで突っ走って書いてしまったもの。コーディングの意識が変わった今見返してみると、酷いコードだなと自分でも思う。

悪いコーディング例として参考にしてもらいたい。

○ 新規作成

新しい文章を追加する機能。

エントリやコンテナといったデータ構造がよくわかっていなかったが、Mac版文庫番のソースコードを参考にしたり、教えてもらうことでなんとか書けたもの。

はじめのコードでは、新規作成した文書のタイトルや本文にゴミ(文字列)が入るというバグが起きていたが、これはメモリを初期化するのを忘れていたからであった。メモリの使い方に関してはインターン期間中の仕事で嫌というほど苦しめられた。

○ ノート保存

本文保存と比べると、RTF 形式での保存をしない(コールバック関数を使わない)ので楽にプログラムを組めるかなと思ったが、今度は文字コードをSJISからUTF 8に変換して保存するという慣れない方法に苦しめられた。

文字コードを変換するとバイト数が変化するので、このことを考慮してメモリを適切に確保しなければならない。ここでは特にNULLポインタ分の1バイトをどこでどう確保するか悩まされた。

学校のプログラミング課題でもメモリ確保に関するエラーはよくおこっていたが、ここまでメモリに関するエラーに悩まされたのは初めてだった。最終的に久永さんにメモリ確保の手順を覚えてもらうことで解決できた。

今のところ不具合は見つかっていない。

○ ノート読み込み

ノート保存の機能ができたと思ってテストをしてみるが、文章を開いてみてもノート内が空になってしまう事態が発生した。

「どうしてこのプログラムでダメなんだ?! 保存はできていないのか! ?」と悩んでいると、猪狩さんがログをとってプログラムの動作を確認する方法を教えてくれた。

ログをとってみると保存はできているようだ。ということは…と考え見てみると、案の定ノートデータの読み込みの方ができていなかった。

こういうムダな時間をなくすために次の方法が効果的であることを学んだ。

関数の開始部分と終了部分やIF文の中などに、そのコードを通ったことがわかるようなログを出力することで、ある操作に対して意図したとおりに動いているかをチェックする。

○ フォント変更

本文のフォントを変更できるようにした。メニュー項目から変更できるフォントは次のもの。

- ボールド
- イタリック
- 下線

今のところボールドだけは、カレット直前の文字がボールドであるときにチェックマークが入るようになっている。他の項目はフォント変更の作業中に白石さんから「本文保存で文庫番が落ちる」という報告があり、本文保存のテストと修正に作業を切り替えたという理由で未実装である。ついつい機能を追加するのが楽しく、不具合がかくれたままで進めちゃうというNG行為であった。

フォントパネルからフォントを選ぶことで変更できるフォントは次のもの。

- フォント名 (MS ゴシック, MS 明朝体など)
- ボールド
- イタリック
- 文字サイズ
- 色
- 打消し線
- 下線
- 文字セット (日本語, 欧文, ギリシャ語など)

カレット直前の文字のフォントを取得して、そのフォント情報をパネルの初期値として設定する機能ができている。

実装方法を調べるのに時間はかかったが、実装は本文保存やノート保存のように悩むことなくできた。

ただ動作テストをしたときに、文字を「MS ゴシック」体にすると日本語が縦書きになってしまうという現象がおこり、どうしてそうなるのか悩んでいた。このままではまたしてもムダに時間を浪費してしまうと考えていたとき、ちょうど猪狩さんがどんな機能を実装しているのかと尋ねてきたので、実演してみせた。すると、「これ縦書き用のフォントじゃん」という答えをいただいた。…確かに「MS ゴシック」ではなく縦書き用のフォント「@MS ゴシック」を選んでいた…。

こんなくだらないことを早く発見するためにも、他人に訊くことは大事だと実感した。

のちにフォントパネルを表示してフォントを変更する部分は、久永さんに「コードがひどい」と言われ、修正してもらった。危うく悪いコーディング例をまたひとつ残してしまうところであった。

モジュール化の具体例をここで学んだ。

○ 【文字】->【テキスト】

つぎのメニュー項目の機能を実装した。

- 左揃え
- 右揃え
- 中央揃え

自力で実装できた機能である。インターン期間中に不具合は見つかっていない。

○ 不具合について

不具合を見つけるために動作テストはこまめにしなければならない。

不具合がある状態のまま次の機能を実装してしまうと原因の特定が難しい。

実際、機能が増えていくとコード量が増えていくので、修正すべきところを探すのが面倒になる。

機能を次々に実装していくと楽しいが、ひとつひとつの機能に不具合が潜んでいるような仕事をしてはいけない。ひとつの機能を実装したら、不具合が見つからなくなるまでテストをし、完璧になったら次の機能実装へ進むべきであった。

○ リリース

ソフトの機能実装にともない実装内容や不具合があればその減少を詳しく記したメールを、メーリングリストに送信する作業。自分が最も苦しんだ作業である。

「誰が読んでもわかるような文章で」書くということがとても難しく、文章の組み立て方や単語の選び方に苦労した。

はじめのころのリリース内容はとてもひどいものとなっている。悪いリリース例として役立ててもらいたい。
自分としてはかなり時間をかけてしまった作業なので、コーディング規約のようにリリース規約もあったほうが良いと思った。
リリースを書くときもプログラムを書くときと同じで、誰かに読んでもらってわかりにくいところを指摘してもらうことが必要である。毎日何度も読んでくださった久永さん、ありがとうございました。

感想

人は成功よりも失敗を経て学ぶことが多いというが、まさにその通りであった。というより仕事で成功した覚えがそんなにないので、より強く「やってはいけないこと」、「やらなければいけないこと」を学べた。正式に社会に出るまえにたくさんの失敗ができたことはとても良い経験になった。

社会の常識やプロのやり方を教えてもらったが、なかでも特にショックだったことは、学校と会社との圧倒的な違いだ。
今までは学校で習ったことはそのまま会社で使うものだと思い込んでいたが、そうではないということを思い知らされた。このことはこれからの学校生活で常に意識していきたい。

治郎吉で学んだことは、絶対これからの学校生活、就職活動、就職後に活用していけると思う。毎年ボーっと過ごしていた夏休みだが、今年は自分の大きな財産ができた。