会　議　議　事　録　（抄）

会　議　名	2025 年度専門学校東京テクニカルカレッジ 第一回情報処理科教育課程編成委員会
開 催 日 時	2025 年 7 月 18 日（金）15 時 40 分～17 時 00 分
会　　場	専門学校東京テクニカルカレッジ 地下 1 階 テラホール
参　加　者	＜外部委員＞ 後藤　英明　（株式会社ネクストワン） 開原　裕一　（株式会社ネクストワン） 経塚　真裕　（ソフトヒューベリオン株式会社） 片山　善久　（株式会社エフ・エム） 坂上　誠　　（株式会社テクノプロ　テクノプロデザイン社） ＜内部委員＞ 井坂　昭司（専門学校東京テクニカルカレッジ　情報処理科科長） 呉石　義明（専門学校東京テクニカルカレッジ　情報処理科主任）
	＜系別分科会＞（第二部） 1.議長挨拶　井坂より挨拶 2.前回（系別分科会）議事録の確認 3.意見交換　IT 業界の変化、生成 AI の活用状況、 ソフトウェアのバージョン管理について、その他
討 議 内 容	**１＆２.IT 業界の変化と生成 AI の活用状況** （坂上） 我々は製造業のお客様なので、開発ツールとかプログラミングをしていく中で、まだまだ AI を使ってどうのこうのという段階ではない。製造業のお客様はすごくポジティブなわけではなく、何かあったときの説明責任が、AI がブラックボックスになるようであればできない。なので、制御とかも含めて AI でやるのかという空気は去年くらいまではあった気はする。今年くらいから、そんなことを言っているのは日本だけでは？という感覚がお客さんからも出てきている。それを共同で研究しないかといった我々に対する問いかけみたいなものがポロポロと出始めている。 （片山） 少しメーカーさんたちの動きがある。最近インフラの案件が極端に減っている。ただ業界的に言うと、そんなに開発は減っていないのに、なぜその案件が出てこないかを考えると、各企業のパートナー会社に行くと、クラウドシフトと言ってアプリの人たちをインフラに回している。なので、各メーカーのインフラを内製化しているようで、外側に出てこない、仕事が出てこないということが起きている。アプリケーションは AI で効率化できるということがだんだんわかってきたので、アプリの要員がそれほどいらないのではということ。どちらかというとインフラ側にメーカーさんたちが力を入れているというのを非常に感じた半年間だった。今うちの会社もインフラがちょっと苦戦している。仕事がない。 （経塚） 僕も人材不足と言われている割には、そこまで感じていない。2030 年までに 40 万人くらい足らなくなるといった推計が出ているが、実際のところこんな仕事も増えているとか、人手が足りないというよりは、大規模案件は少なくなってきているような気がする。 生成 AI は前回お話したよりはあまり効果がないと感じている。 プロンプトエンジニアとしてどうコミュニケーションをプロンプトで取っていくか。弊社としてはプロンプトのやり方を教える研修を取り入れて、会社としても社内議事録は全部生成 AI に読み込ませて、例えば言葉だけで終わった内記とかって前回どうなっているのかというのは、生成 AI に全て記憶させた。 アイデアやデザインとかは、今まで外注していたのを効率化することで経費削減の流れになっている。あとはコールセンター。どんどん歴を読み込ませて同じようにやるのは人材

	を縮小させるにはいいが、半年、1 年前もあったので、大手も含めて生成 AI がそれ以上にもっとすごい段階に入ったかというとまだ感じていない。大手が設計から全部 AI を使っていくといった成功事例が出たり、今まで例えば 10 億円かかっていたのが 3 億円でできたという結果が出たりというのはまだない。もう少ししたら出てくるのかもしれないが、明らかにそこまで業界が変わったかと言われると、まだ感じてはいない。 （井坂） 開発現場もそこまで AI を使って開発しているというわけではない? （経塚） そうですね。逆に言うと個人に任せて要領よくやったり。特に金融は外部的に情報を出してはいけないという話があるので、そこも制御しながら、顧客の OK が出ない限りは多分使えない。僕はもう少し飛び越えて、じゃあ生成 AI だけでいいのかというふうに見ないとと思っている。日本は生成 AI を使いこなせるようになったが、そうではなく、それをやっている以上世界と戦えないのではと思う。日本独自のものを作っていかないと怖いと感じている。 （開原） 当社に関してはインフラ側なので、変化があるとすると、オンプレの環境プラス Azure、AWS でやろうというのが増えている傾向にはある。仕事が少なくなっているとか、人が削減されているかというと、それはない。 リプレイスの時期には稼働も当然高まるし、本当に完全なオンプレからクラウドに移行というのはまだ出ていない。その一方で開発系は、例えば JR さんとか次世代の券売機を作ろうという動きにもなっているので、生成 AI が入ってきたからとか、何かが理由でこの仕事が減ってきた、需要がない、といったことはないと思う。 逆にいろんな SIer の企業が新しいサービスをリリースするので、そのサービスに則った知識、例えば元々通信インフラの世界の人間が突然 ISP の方の仕事に携わったりとか、 そこの業界独自の新しい提案内容に合わせる形のエンジニアを作っていかないと、対応ができなくなるというのはこの 1 年くらいのかんじ。 Sler が求める動きを追えるような知識をある程度入れておかないと、当然単純にコーディングやカラムの設計ができるから OK というわけではなく、そこにプラスした知識がないと、この先仕事をしていくのがどんどん高度化していく。一方で AI でというと、領収書とか、タクシー券とか接待費で使ったのが、社内の経費で使ったのかわからないこともあるので、 一概に経理の職がなくなることはなく、うまく活用して最終的に人間がジャッジするという部分は、ここ一年であまり変化はない。 当然コーディングの中身を精査するというような使い方もできると思うが、最終的にジャッジするのは必ず人だと思うので、ブロックとの対話の仕方であったり、それぞれの AI が持っているデータベースがどこが発祥だったのかというのを遡って、どんな機械で学習が進んでいったのかというのを知っておかないと、うまく自分の求めているものに対しての応答が返ってこないことがある。そういったグレーゾーンの部分を逆に知れるような深みのある人材がこれから求められてくると思う。 （井坂） 一通り聞かせていただいたが、基本的には我々の今のカリキュラムが、 ちゃんとプログラムをしっかり作れるようにするためのベースは崩す必要はないと思う。2 年過程の中で、どのくらいまで AI の学習を学生にさせるか、プロンプトのこれくらいまではできるようにさせておいた方がいいとか、そんなご意見がいただけるとありがたい。新しいカリキュラムというか。 （経塚） 今は使い方だけの研修をやっている。今年の研修は、生成 AI を使ってこの成果物を作るというコンテストをやらせようかと思っています。コンテストで、優勝したら何かあげるとか。学生だと何かあげるというのは難しいかもしれないが、そうすると生成 AI を使いこなしたいエンジニアたちは燃えるので、それで動機付けする。 デザインとかアイデアとか、生成 AI で全部できるらしい。今までは 1、2 ヶ月かかってデ

	ザイナーに頼んでいたものが、1 日 2 日で 10 種類くらい作れるので、一番近いものをデザインナーと話し合う。1 つのカンフル剤になるらしい。絵や動画、アニメも作れる。絵を見せて、これをイメージしたポップの音楽とか、演歌を作れるという話もあるので、もう著作権を超えてくるという話。初めてアウトプットで面白いものが出てきたら、こういうこともできるね、ああいうこともできるねというのが始まるのかなと。 （開原） 実際に SU-NO というアプリがある。例えば、自作の歌詞をある程度イメージしたものを入れると、1 分くらいで曲を作ってくれる。そういうのを使ってグループディスカッションさせた企画は、通常の言語学習とかではない別枠で、こんなことが今できるというので、興味関心を沸かせて、その上でどんなガイドラインがあるのかというような、一つのアプリケーションを使っていくというのも面白かった。 研修でサーバネットワークや Windows サーバをやるが、プライベートな時間に AI を触ったことがあるかと聞くと、知ってはいるがチャット GPT くらいしか使ったことがないと言う。どこまで進んでいるかなど、知らない部分が多いので、そこを認知させて、正しいかそうでないか、どこまで使っていいのかというガイドラインを自分たちで探すということを、カリキュラムの一環で取り入れたのは、ちょうど 1 年前くらい。 （井坂） 我々も教育機関なので、どのタイミングでどう教えるか。呉石が課題を出すと、AI でやってくる。それはまたひどい状態になっていたりする。だからちゃんと AI に頼らず、勉強して理解してもらうために課題を出すが、それが学生からすれば、課題を出すことが目的になっている。 （呉石） 使い方も Google の検索の延長にしかなっていない。例えば絶対に書けないレベルのコードを出してくるが、良し悪しの判定を本人はできないので、こちらは一発でわかる。 他のもそう。会社に出す履歴書のコンテンツも生成 AI で作っている学生がいるはず。言葉の使い回しのチェックとか、日本人の学生でもそのチェックをする能力はない。 （片山） 新入社員研修で、最初からスムーズに進みすぎていて、何かおかしいと思い調べたら、今話に出ていた通りで、コードに対して質問をしたら全く答えられない。ほとんどの子たちがチャット GPT から返ってきた回答を組み合わせて、プログラムを書いていた。その時点から生成 AI を禁止にした。その途端にペースががた落ちになった。そこでやはりジレンマがあり、今からの世の中は生成 AI を使いこなすということもあるけれど、それを使ったせいで自分の能力が伸びないことがいいことなのか、教育部で議論している。 中にはやはり使ってもいいのでは、作ったものに対してきちんと答えられればいいのではと言う人もいるが、そこまでできない子たちは全然能力がつかないので今回は全面禁止にした。 （井坂） うちの今の学生も多分そう。だから履修判定試験をやると、え?という結果になってくる。だから結構頼っていると思う。 （呉石） 一昔前は Google とかで検索すると、サンプルコードが出てきて、変数名や呼び出し方、ライブラリの名前を変える等、最低限それくらいはできないといけなかった。今は使っているものを指定すれば、綺麗なコードを出してくれる。課題を出してもらうと、成果物としてはよくできている。だから個別で口頭試問するしかない。なんとかして時間を作るが、それも手が回らなくなったら、あとはお祈りしながら試験をやってもらうしかない。 （井坂） 本当に誰がどれだけできているのかというのは、途中の段階で把握ができない。教育現場も本当にそれをどう見極めるか、阻止するか、それは難しいところ。

	（片山） 新入社員に対しては使用禁止としたが、社内では使っていかないとというジレンマがある。これは自分の会社のカリキュラム。生成 AI の入門があり、それを活用するという部分があり、ラグの環境構築、ラグというのはデータをクローズの中で読み込ませる環境を作るということ。それに対して生成アプリケーションの開発演習で、プログラムに生成 AI を組み込み、使えるようにするという講習と、実際にインフラとかで障害分析を行い、ログ等を読み込ませることで、生成 AI に障害を判断させて解決策まで出す。こういう形で生成 AI を使いこなせるところまで持っていこうとカリキュラムを作った。 チャット GPT に基づく活用編までは社員全員。生成 AI の危険性や、使用の良し悪しを教えないといけないので、生成 AI のガイドラインと合わせてこの 2 つは社員教育としてやっている。ラグのところからは、実際に必要な人たち、アプリケーション開発のメンバーに特化している。障害分析ソリューションは、インフラのメンバーも入れている。少しくらいは社員に触れさせて、生成 AI は何者か分かるように進めている。 （経塚） 黙っていてもあと 3 年も 5 年もしたら誰でも使えるようになる。 使い方というより、基礎が理解しているかを解いていくべきだと思う。私たちの会社も研修期間中、生成 AI を使っていいかと言われたとき、どちらがいいと思うのかとしか聞かない。ダメと言って終わり。今回、本当にできているか疑わしい子が何人か出たので、30 人くらいの新入社員を 15 人のペアにしてバグの出し合いをさせた。生成 AI を使わずに直せと言ったら、1 週間かかる人と 2 日で直す人がいて、2 日で直す子はちゃんとやっていた子だった。1 週間かかった子もそこで力をつけられたので、それは良かったと思う。 （井坂） 不思議に思うのは、生成 AI を使っても自分のためにならない、技術を学びにきているのに、それでいいのか淡々とそういう話をして、これだけ言えば自分でやるだろうと思っても、やらない。今のお話を聞くと、社会人になってもやらないのでは。考えれば自分が絶対損をするとわかるのに、どうしてやるのか。 （呉石） そこは視点の違いで、とりあえず目の前の今日の課題が終わればいいと思っている。 出したもののおかしさを自分で判定できないから出してくる。それはそれで教材になるので、例えばここの変数名が違う人は、変数の定義のところはプロンプトで読ませてやりなさいとか、みんなで共有する。あとは、これを僕が指摘できて、君たちができないのは、僕がその分勉強しているからであって、君たちには出力されたものの良し悪しが判定できないからこのままでいったら事故るよという話をして、半分くらいの学生は聞いてくれる。 （坂上） 仕事に対する面白さが分かれば自分でやると思う。昨日九州の大学の先生と話をした。機械工学科の先生で、その人はどちらかと言うと先進的な人なので、学生に卒論に何を使ってもいいとしていた。機械工学の子でもアウトプットをよくしようと思ったら、プログラミングを自分でやらないといけない。でもプログラミングは自分でできないから、生成 AI を使って勝手にやる。そんな世界。生成 AI ネイティブが 3 年後に来るのでそこはもう止めても仕方がない。だからもう使えと言っている。さらにソフトウェアのエンジニアとしては、面白いことを伝えてあげられれば、両方使いこなす人になる。それが一番いい。使うのは当たり前だと思う。機械、電気、電子関係なく、バイオの人も関係なく絶対使う。全員が最低限プログラミングできて当たり前になるので。 （井坂） 面白いのは、我々もそんなにがつがつ生成 AI を使っている方ではないが、うちの自動車の先生なんかすごい、IT 系でない人の方が結構面白がって使っている。

（坂上）
そこだと思う。機械とかの人は生成 AI でプログラミングができるようになったと思い面白くて、どんどん学んで使う。VS とか知らないですか。AI 同士で 同じものを作るのに AI しか使わない。同じアウトプットを求めさせて、コーディング一切禁止。やる中で面白さが出てくるのでは。

（井坂）
2 年生の後半くらいになれば、そういう制作で開発系いかない子たちが総員でちょっと作ってみるとか、あり得ますね。

（坂上）
最近はエンジニアに向いていないと退職する若手が増えてきた。面白くないのだろうと思う。IT やっていると食べていけるみたいな感じで入った子に多い印象。今面接のときの過去データとかを分析し始めた。多分本質的にエンジニアをやりたかった子ではない。

（井坂）
よく大学生の 30 何パーセントが IT 系に就職するっていう話があって、また 3 年以内の離職率が 30 何パーセントと言われる。

（片山）
うちは 5 年以内の離職率は 1%くらい。採用面接のときに学生の 99%がアプリケーションとかプログラムで世の中のためになりたいと言ってうちの会社に入ってくる。
入社後の研修をやり、最後どこの部に配属するかという面接をやると、半分の子たちは自らプログラムを作るのを諦めている。自分でも分かっていて、書けない子は、書ける子を見るとその差が歴然と見えてしまうので、だんだん嫌になる。だからアプリケーションに行きたい子とそうでない子と別れる。そこから、サーバーエンジニアやソフトワークエンジニアやデータベースエンジニアがあるという話をして進路選択をするので、うちは辞めないのではないかと思う。

（井坂）
意外とそう。うちの学生も、みんな一応は開発を目指してくるが、やっているうちに自分の中ではもう無理だということがわかる。そうなると、就職する段階で開発にはいかない。

3. ソフトウェアのバージョン管理について

（井坂）
うちは Java にしろ Oracle にしろ バージョンアップがあっても最先端のバージョンを使えるわけではない。各企業さんはどうバージョンを管理しているのか。

（片山）
お客さんによって、最新のものを使いたいところと、実績のあるものを使いたいところで違う。銀行とかは意外と障害が起きるので、前のバージョンを使うとか、世の中で安定していると言われるバージョンを使いたいと言われる。バージョンは各現場でマチマチ。

（井坂）
仕事によりバージョンが変わるから、それにプログラムを合わせるしかない。

（片山）
あとは EOSL をソフトが迎えてくるので、サポート切れが起きるとどうしても上げなければならない。そのときも一番上に上げるのか、一個前くらいにするのか、それも企業による。ある程度リスクも含めてこのバージョンだと次の EOSL を迎えるのがいつと提示して、どこを選ぶかという形になる。その間が空きすぎると、一回でバージョンアップできないということもある。ミドルウェアは、そのデータも引き継がなければならない。だからその一個前からは移行できるが、二つあくと、データの形を変えないとダメということも起きる。

	なので一個一個上げておかないといけないソフトもある。例えばファイル転送ソフトとか二つ飛びとかにすると、設定を変えなければならなくて時間がかかるということも起きる。 （経塚） 今思っているのは、これこそ生成 AI に違いを聞いたほうが簡単にできる。生成 AI に Oracle これとか、Windows これとか入れて前バージョンと何が違うのかを聞く。 （開原） クラウドの仕事をしていたことがあり、その時は逆にダウングレードをして、直にやり取りして期間延長をしていた。結局バージョンが上がると、検証作業に半年かかるので、そこまでには上げなければならない。そこを超えた分は有償でいいから昔のバージョンのものを納品してくれというところもある。最新が一番いいという業界もあれば、どうしても検証が済んでいるものでないと配信ができないというので結構分かれたりする。 バージョン管理は最新のものが出たら、それが適用されるのは 1 年後とか。そこでまた上げるという形で、実際にうちの会社の端末なんかはスペックを指定して、全部会社のほうで、オーダーをかけてその通りのものしか納品させないようにしている。それくらいしないと、勝手にバージョンが上がると下に入れているツールが動かなくなったりする。そこの検証期間は必ず設けなければいけない。銀行さんとかはやはり古いもので、検証も完全に終わり、テストも問題なく、実装できるのは 2 年先とかが実情。 （井坂） あと学生の中であったのはコーディング規約。コーディング規約もその仕事ごとで変わっていく。スペースを空ける？ 一つの i イコール i プラス 1 のスペースを空けるとか、そういうところはどうなのか。 （開原） いや、規約に入ってない。どっちでもいいと思う。 （経塚） 多分そこは個性では。縛られないところだと思う。逆に名前の付け方とか命名規則。 （呉石） プロジェクト単位で決める時もある。コメントの付け方にしても。 そういうのはチームごとに決めたりする? （経塚） そういう違いは、会社さんの歴史。Sler とかメーカーの歴史でどこかから絶対に持ってくる。それをプロジェクトで少し改変するくらい。変わるとしたら、そこの開発は大手 Sler とかメーカーとかが委託される前に、エンドユーザーの情報システムが介入してくる時。例えば勝手に日本語にしたり、英名にしたかったのにデータベース名を全部日本名にされていたりとかはよくある。お客さんがそうでないと嫌だと。納品するときに分からないとダメだという話。プロジェクトごとというのは、情報システム部さんがどれくらい介入してくるかによると思う。 （呉石） 学校は今年から Java のバージョンを 21 にした。 その前は 11 だったが、学校の場合だと市販で教科書のテキストのようなものがあるかも結構大事。あと学校で入れる第二の条件は、サポート期間が長いもの。Java だと 3 つか 4 つに一回ロングサポートがあるので、基本それでないといけない。今 17 のほうが出だしたくらい。 Web アプリのほうのライブラリのパッケージ名が全部変わるので、テキストを全部編集し直して、動くかどうか検証中。先に学生のほうだけ全部変えたので、配布予定のライブラリがそのバージョンで動くのかというのを検証しながら、テキストの直しをしている。

	４．その他 （坂上） 質問でもいいですか?今日、色々と学校の取り組みを見て、違う学科同士でコラボしてすごく楽しそうだと思った。その中で、情報処理チームのコラボ例が多くなかったのは何か意図的なのか。 （井坂） いや特に、それぞれ問題発見があった段階でこの学科とこれで解決できるかみたいな話になるので。なかなかそういう案件がうちの学科にはないというところ。 今 IoT でなんでも片付けられる。さっきのバイオコロニーの菌の数を AI で数えるとか、画像処理で、そういうものがやりやすい。 （坂上） そういうのがあれば、さっきの面白さみたいな話ではないが。 学生が中身を見てものが動くところを見るとか、手触り感のあるものとか、社会で仕事をするとなればいろんなチームと関わることになり、ソフトウェアのエンジニアだけで何かをするということはなくなってくる。 だから、この情報処理科の方々もそういうものを取り組んでいけるといいなと思って見ていた。 （井坂） 情報処理科としては、メインは 1 年生の最後に掲示板を作り、2 年生の最後にもう少し発展した SNS やショッピングサイトなど、そういうところを学科の中で作っている。 その辺が中心で、外部の学科というのはなかなかテーマが難しい部分もある。どっちかというとこっちが主導じゃない部分が多くて、こういうことを IT で解決してくれないかという方が多くなってくる。そういう要求がバイオや環境などからは出てくるので、そっちの学科の方がどうしても多くなってしまうというのが現状。 （呉石） センサーとかツールとかの値段がどんどん下がってきて、普通の学校の教材としても扱えるものが増えてきている。データサイエンスとか IoT とかでやれることがすごく増えている。 （井坂） うちも RJP で IoT チックなことをやっている。 今回の資料の中だとエサやり機。 今年オープンキャンパスでドローンにプログラムを埋め込んだ。結構学生は楽しくやる。 今年は RJP の中でドローンを情報処理科でも買って、もう少し楽しませる要素をいれようかと考えている。 （経塚） ドローンの花火すごく綺麗。 大阪万博とかでもやっている。 （井坂） そういうプログラムを入れて仕組みが分かれば、 同期させれば同じ動きをするし。 あとメイン機器を置いておいて、そこから自分はどうやって離れていくのかみたいなことをやれば そういう花火みたいなものもできるかもしれない。そのきっかけみたいなところが面白いかなと思う。 値段も 19,800 円くらいで買えるので、それを買って遊ばせようかなと思っている。 以　上