JUAS アドバンスト研究会
新卒 3 年でフルスタックエンジニアを育てる

最終報告書

2018 年 3 月 31 日

目次

1 はじめに

1.1 本研究内容を選定した背景

近年、企業のITに対する期待が、コスト削減から事業拡大へ変わりつつある。従来、ITに対する期待は、業務の効率化やコスト削減、マニュアル作業の自動化であり、かつ構築するシステムは高品質で高信頼性を求めていた。

そのため、企業（経営者）が重視するのは、効率性や生産性、投資対効果（ROI）であり、社内顧客であるユーザ部門の満足を得ることが重要であった。

そこでシステム開発現場においても、ユーザ部門との要件確認が重要になり、時間をかけて要件定義を実施していた。ここで取り決めた要件に従ってシステム開発を行うため、前工程の成果物の品質を確保し、前工程への後戻り（手戻り）を最小限にするウォーターフォールモデルでの開発が現在の主流な開発モデルとなっている。このウォーターフォールモデルを得意とするベンダーを適材適所に配置し、ユーザ部門が満足するシステムを作り上げてきた。

しかし近年では、「デジタル化」という言葉に代表されるようにITの活用方法が大きく変化し、他社との差別化だけにとどまらず、既存のビジネスモデルを破壊する力を持つようになってきている。

デジタル化をいち早く成し遂げた企業が、業界の垣根を超え、短期で市場を席捲（破壊的イノベーション）している。つまり、業務効率化やコスト削減のツールではなく、新たな市場の獲得のためにIT利用が本格化している。それを受けてシステム開発現場においても働き方が変わってきている。

企業におけるシステム開発現場において、これまでの適材適所の外注開発から内製化に変わり、技術力・専門性を自社内に蓄積するようになってきている。このため、IT部門社員にはインフラ構築からフロントアプリ開発まで（フルスタック）実装する技術力が必要となった。このような事業環境の変化において、各企業はフルスタックエンジニアを必要としているが、世間的にフルスタックエンジニアが枯渇しているのが現状である。

そのためIT部門の新入社員をフルスタックエンジニアに育成することが各企業のミッションとなっている。

このような市場の変化に伴い、ITの変化からエンジニアに求める技術の変化が起きている状況下において、フルスタックエンジニアを育成するために必要な手段、方法、環境を本研究会にて研究するものである。

2 フルスタックエンジニア

「フルスタックエンジニア」と一言で言っても、職種や役職、業種などそれぞれの立場によって定義が異なる。

そこで本研究会では、それぞれの立場の観点から「フルスタックエンジニア」を定義することとした。

各研究員の考えるフルスタックエンジニアを別紙に記載する。

2018 年 3 月 31 日

JUAS アドバンスド研究会

「新卒 3 年でフルスタックエンジニアを育てる」

1. フルスタックエンジニアの定義

数年ほど前から IT 専門誌や Web サイトなどで「フルスタックエンジニアが必要だ」とか、「フルスタックエンジニアを目指すべきだ」などという言葉を目にすることが多くなった。実際、普段何気ない会話の中で「フルスタックエンジニア」という言葉を違和感なく使っている。

一方、このフルスタックエンジニアを育成しようという話になると話が変わってくる。何故ならば、このフルスタックエンジニアという人材に対するイメージが人によってそれぞれ異なるからだ。

日常会話の一部として使う「フルスタックエンジニア」という言葉は、おそらく多くのエンジニアの間において、大まかな概念としての認識は一致している。しかしその言葉に対して、ひとたび「育成」という具体的なテーマを与えた瞬間、会話が成立しなくなる。これは、それまで一致していた「フルスタックエンジニア」という概念を具現化するあたり、より具体的なイメージが必要となり、それが、エンジニア間でそれぞれ異なるためだ。

そこで本章では先ずフルスタックエンジニアの定義について考えたい。

一般的な定義

新しい言葉を調べるときに、真っ先に思いつく先はインターネットであり、wiki ペディアであろう。そこで wiki ペディアでフルスタックエンジニアを検索したところ結果はゼロ件であった。Wiki ペディアが全てではないが、ここに記載されていないということは、世間一般ではまだ浸透していないか、または、定義が曖昧な言葉なのかもしれない。

GGRKS (注 1) という言葉にあるように、次に Google で検索してみたところ 1,060,000 件もヒットした。以下に、検索結果上位で述べているフルスタックエンジニアの定義について紹介する。

フルスタックエンジニアは、マルチエンジニアとも言われており、通常ではシステムエンジニア、サーバーエンジニア、データベースエンジニア、ネットワークエンジニアなど、専門の技術者が分業してプロジェクトを進めるのに対し、フルスタックエンジニアはすべて一人で開発を行います。

（中略）
フルスタックエンジニアには、ソフトウェアやデザイン、およびサーバーなどの開発に関することから、ネットワーク・セキュリティ・データベースまで、IT に関する幅広い知識が要求されます。

出典：CAREER HACK
http://careerhack.en-japan.com/glossary/fullstack-engineer

フルスタックエンジニアと言っても色々定義があるようだ。私の認識する範囲で言うなら、要は何でもできる IT エンジニアだと考えている。DB とかサーバ、ネットワーク、要件定義から開発までまとめて面倒を見ることのできるエンジニアだ。現場にいたらヒーローだろう。

出典：エンジニアライフ
http://el.jibun.atmarkit.co.jp/101sini/2017/04/post_40.html

そもそもフルスタックエンジニアという言葉に明確な定義がありませんが、一般的には“web サイトからアプリまで、一貫した複数のスキルを持っているエンジニア”というのが共通認識となります。

出典：IT プロパートナーズ
https://itpropartners.com/blog/7311/

フルスタックエンジニアとは、特別に定義された言葉ではなく自然発生的に出てきたものなので、線引きはあいまいだが、一般的にはクラウド等でのインフラ構築から、アプリケーションの開発、簡単なフロントエンドの実装までを 1 人で担当できる「Web アプリケーションを 1 人で作りきることのできるエンジニア」を指すことが多い。

出典：日経ビジネス　ON LINE
http://business.nikkeibp.co.jp/article/report/20150107/275927/

これらインターネット上に公開されている定義を読んで気がつくことがある。それは、「フルスタックエンジニア」という言葉には、未だ明確な定義が存在していないということだ。

各サイトで共通して言えることは、どのサイトでもフルスタックエンジニアについて、二つの軸で語られている点である。
一つは「要件定義」「設計」「実装」と言った、開発プロセスに関する軸であり、もう一つは、ハードウェア・ミドルウェア・アプリケーションと言ったシステムアーキテクチャ階層に関する軸だ。

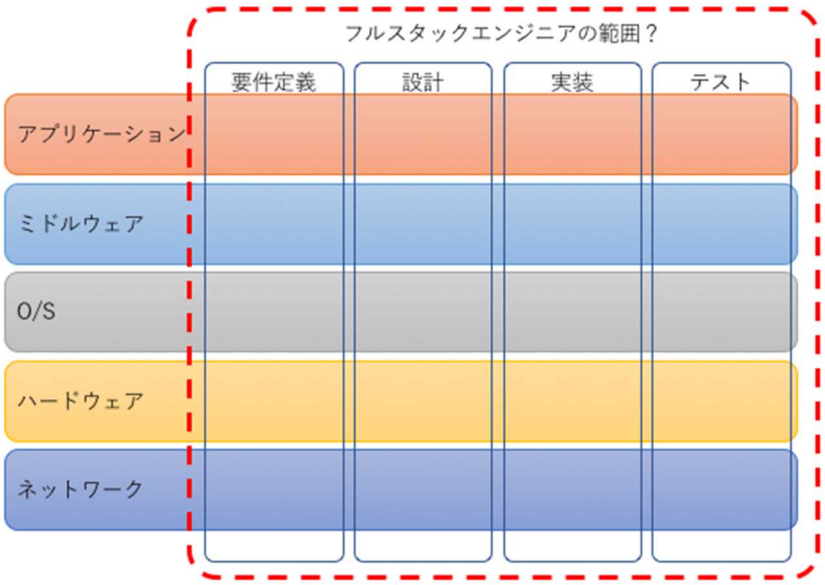


図１：フルスタックエンジニアの範囲

しかし考えてほしい。どの記事も定義や線引きは曖昧としながらも、図１の赤線で囲った範囲全てを一人で行うのがフルスタックエンジニアだとしている。もしそうであるならば、フルスタックエンジニアに求められるスキルは広範囲となる。そもそも、フルスタックエンジニアは存在するのか？という疑問さえ湧いてくる。

　アプリケーション開発を例に考えれば、どこか特定フェーズ、例えば「実装」に関するスキルを習得するだけでもかなりの努力を要する。にもかかわらず、この定義で行くと、フルスタックエンジニアになるには、全てのフェーズで、かつ全てのレイヤーに渡ってある程度の知識を習得しなければならない。

これは容易なことではない。ましてや新卒レベルのエンジニアを、わずか 3 年で育成できる範囲など、たかが知れているのではないかとさえ考えてしまう。

企業が期待する人材像

そもそも企業はフルスタックエンジニアに何を期待しているのであろうか？企業とて何も考えないわけではない。前述のようなスーパーマン的なフルスタックエンジニアがそうそう居ないことは十分理解しているはずである。
では何故、フルスタックエンジニアを求めているのだろうか。

フルスタックエンジニアを定義するということは、企業が求める人材像を明らかにすることと同義ではないかという仮定に行き着く。すなわち、企業が求める人材像を明確にしなければ、フルスタックエンジニアを育成しようにも、そのゴール設定ができていないことと同義ではいかという仮説だ。

これを裏付ける意見として、「フルスタックエンジニアは無用の長物である」、「フルスタックエンジニアを求めるのではなく、開発者はインフラを考えずに開発に注力すべきだ」という考え方もある。これは、インフラ部分はクラウドサービスに任せ、開発者はプログラム開発に専念するという意味だ。

確かに、フルスタックエンジニアを求める背景としては、アプリケーション開発を担当するエンジニアが自分自身でテスト環境の構築やパラメータの設定といったハードウェアからミドルウェアまでの構築が出来れば、開発効率が高まる点に基づいている。また、自分自身で出来るのであれば、サーバーリソースの確認などを運用担当者に確認しなくとも済む。

しかしそれはあくまでオンプレミスで開発することを前提とした従来の開発手法における一連の手順である。

一方で、フルスタックエンジニア不要説を唱えている現場は、明らかにこの現場とは異なる。すなわち、システム開発においてクラウド活用を前提とした現場なのだ。

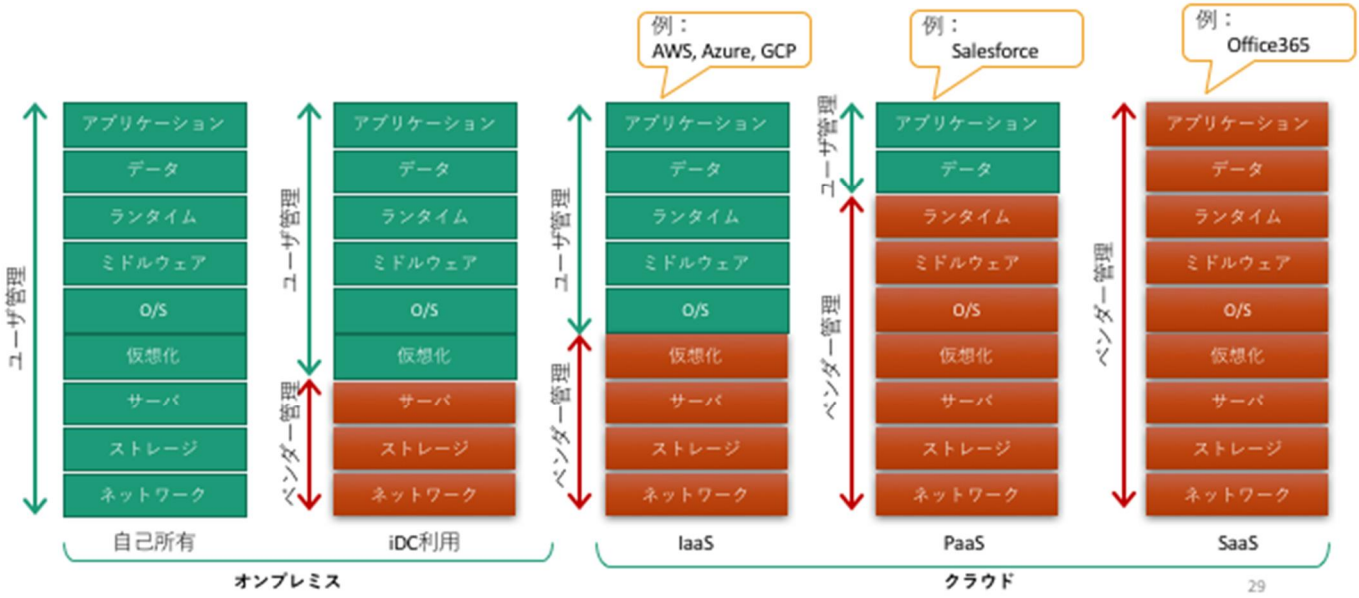


図２：IaaS/PaaS/SaaS の違い

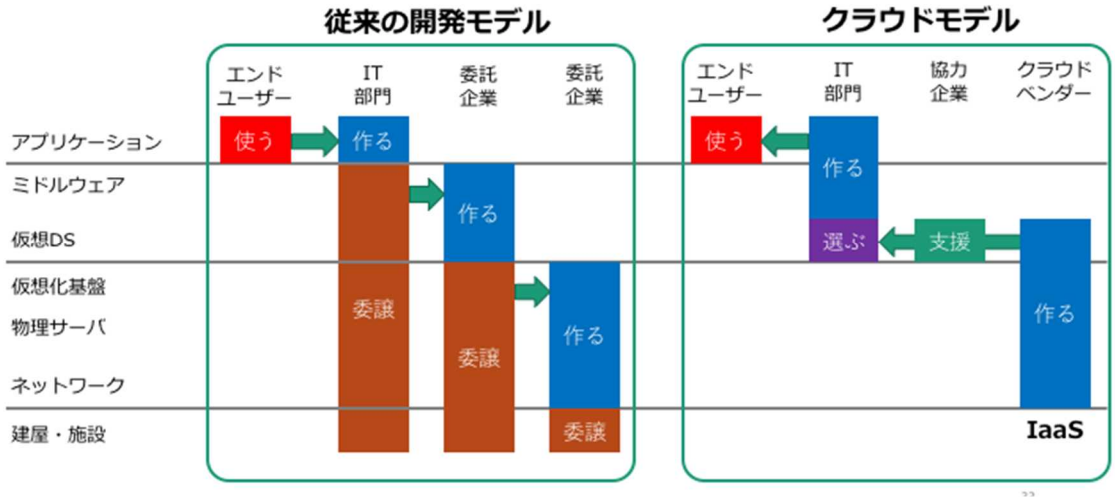
図２から分かるように、オンプレミスとクラウドではエンジニアに求められる守備範囲が明らかに異なる。オンプレミスではネットワークからアプリケーションまでの全てのレイヤーにおいて開発者に管理が委ねられる。（図中ではユーザ管理となっている部分）
ところが、クラウドになると話が違ってくる。IaaS では仮想化まで、PaaS ではランタイムまで、SaaS に至ってはアプリケーションまでもが、クラウドベンダーの管理責任となり基本的に開発者は手出しができない範囲となる。

つまり、クラウド利用を前提とした開発現場では、広く浅く全てのレイヤーをマスターするのではなく、「餅は餅屋」のように、ネットワークやサーバーといった低レイヤーについてはクラウドベンダーにまかせ、開発者はあくまで上位のレイヤーたるアプリケーション開発に注力せよと言っているのだ。

クラウド時代のエンジニア像

それでも、クラウド利用を前提とした現場でも「フルスタックエンジニアが必要だ」と定義する場合には、その求められるスキルは前述オンプレミスの現場で求められるスキルとは大きく異なる。例えば、インフラレイヤーにおいてオンプレミスで求められるのは「構築」に関するスキルだが、クラウド利用前提の現場で求められるスキルは「選択」のスキルとなる。

図３：求められるスキルの変化

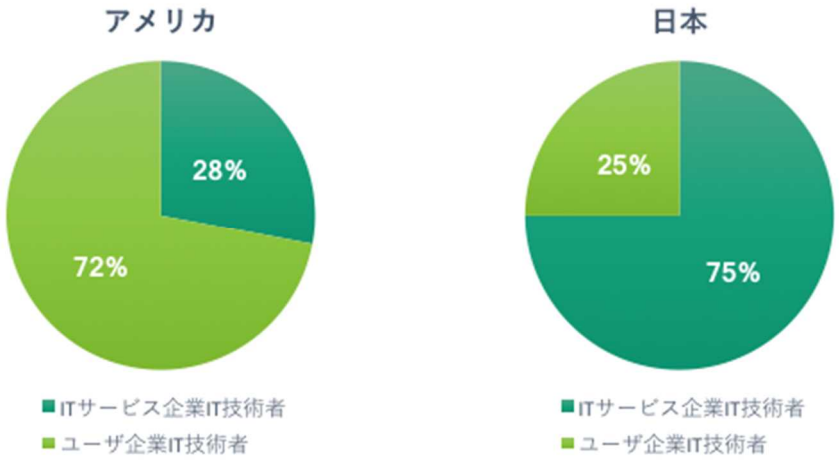


従って、本書のテーマである「フルスタックエンジニア」を定義するには、少なくとも先ず従来のオンプレミスでの現場を想定しているのか、それともクラウド利用開発の現場を想定しているのかを明確にする必要がある。

今回、本研究会でこのテーマを選定した理由の一つとして、新卒を 3 年程度で一人前のエンジニアにしたいという思いがある。しかしながら前述の理由から、新卒をわずか 3 年で従来型オンプレミス開発におけるフルスタックエンジニアに育てるのはかなり厳しいと言わざるを得ない。仮に「育てた」という事例があるとしたら、そのような人材は、そもそも誰の手を借りずとも自分の力でフルスタック領域をマスターできる人材ではないかとさえ疑ってしまう。

それでも育成しようとすれば、それは企業にとって都合の良いフルスタックエンジニアになってしまわないか。もちろん、企業にとって都合の良いフルスタックエンジニアが、世間一般でも通用するフルスタックエンジニアと同義であれば問題ない。しかしながら残念なことに、必ずしもそうであるとは限らないのが現状である。なぜならば、企業によってそのシステム構成は異なり、また企業文化の違いなどによりエンジニアに対する評価の仕組みが異なるからだ。

特に IT エンジニアの７５％以上がベンダー側に在籍していて、欧米のようにユーザ企業側エンジニアが直接開発を行うケースが少ない日本企業において、彼らが求めるフルスタックエンジニアとは何であろうか。



あるときは発注者であるユーザ側エンジニアとして振る舞い、またあるときはアプリケーション開発をマスターしたベンダー的エンジニアとして扱われる便利屋になってしまうのではないか。その便利屋が世間一般でも通用するスキルレベルであれば、本人のモチベーションも保たれるだろうが、その企業でしか通用しないようなスキルだったとしたら、誰が好んでそのようなエンジニアになるだろうか。

こう考えると、ここで述べるフルスタックエンジニアの定義は慎重にならざるを得ない。

本研究会はもともと「10 年後も生き残る SE」の後続として立ち上げた研究会である。そこで本書では、

「企業のためのフルスタックエンジニア」ではなく、その企業で働く「エンジニアのためのフルスタックエンジニア」について提言したい。

器用貧乏

「器用貧乏他宝」という言葉がある。器用な人はいろいろなことができるので人からは便利がられるが、何か一つに徹しないので当人は大成しないが、他人にとっては宝物であるという意味である。「器用貧乏」の同義語としては「多芸は無芸」「なんでもこいに名人無し」「百芸は一芸の精式に如かず」などもあげられる。いずれの言葉も、そう呼ばれた本人からすると、あまり嬉しくない響きであることには違いないだろう。

> 器用貧乏　きよう-びんぼう
> - なまじ起用であるために、あちこちに手を出し、どれも中途半端となって大成しないこと。
> - 器用なために他人から便利がられてこき使われ、自分ではいっこうに大成しないこと。
>
> （出典：Goo 辞書）

フルスタックエンジニアを目指していたのに、いつの間にか企業にこき使われ、気がつくと年を重ねた器用貧乏になってしまっては意味がない。

万能人

一方、「万能人」と呼ばれる人もいる。語源はルネッサンスの人間像で、芸術・技術・科学など、あらゆる分野で才能を発揮する人を指す。その典型として、レオナルド・ダ・ヴィンチやミケランジェロが挙げられる。

フルスタックエンジニアに似た言葉として、トヨタ自動車が掲げた「多能工」がある。
これも、「器用貧乏」ではなく「万能人」を目指したものだろう。

「万能人」と「器用貧乏」、どちらも多彩な能力を持っているにも関わらず、前者は賞賛され、後者はどちらかと言えば蔑まれているように感じる。少なくとも器用貧乏という言葉は、人を褒める時には使わないものだ。では、一体、両者の違いは何であろうか。

エンジニアとして目指すならば、「器用貧乏」なフルスタックエンジニアではなく、「万能人」たるフル

スタックエンジニアを目指したい。

フルスタックエンジニアは勇者

器用貧乏と万能人の違いについて、面白い記事を見つけたので紹介する。

> 私はこう考えます。
> 器用貧乏というのは、万能人の『レベル４０未満』の状態だと。
> 分かりやすく言えば、ドラクエ３の勇者タイプですね。
> レベル４０くらいまでは何かと便利屋扱いされますが、そこから上にいくほど実力を発揮します。
> レベル６０を超えた辺りからは無双状態に突入し、遊び人だけを従えて遊撃に回れるほどの強さを誇ります。
> もしかしたら器用貧乏という言葉は、『万能人』というポテンシャルを秘めた人への妬み嫉みなのかもしれません。
>
> （出典：ジョン曰く https://john-iwaku.com/post-789）

とても面白い考え方である。「ドラゴンクエスト」というゲームを知らない人はピンとこないかもしれないが、知っている人間からするととても的を射ていると感心してしまう。
要するに、器用貧乏も万能人もレベルが違うだけで基本的には同じであると言う主張だ。

筆者はさらにこう続ける。「器用貧乏は中途半端なのではなく、誰しも中途半端に終わることは多々ある。そもそも不器用な人は中途半端にすらなれない。」
その上で、「レベルが低ければ、どのようなタイプであっても優れた才能は発揮できない」と言い切る。

レベル６０を超えた勇者こそが、我々の求めるフルスタックエンジニアであるとするならば、この「レベル」を定義することが同エンジニアの定義を行う上で必須ということになる。

器用貧乏な人の長所としてどのような言葉があるか知人のエンジニア何人かに聞いて見たところ、次のような言葉が出てきた。

・　頭の回転が早い
・　要領がいい
・　飲み込みが早い
・　物事を習得したり覚えたりするのが早い
・　好奇心旺盛
・　切り替えが早い　などなど・・・

「器用貧乏」という言葉自体はネガティブな言葉なのに、その長所となると、賞賛する言葉がたくさん出てくるではないか。
こうして考えると、やはり「器用貧乏」は「万能人」になるための途中段階であると考えて良いだろう。

それでは、どこからが万能人、すなわちレベル６０の勇者なのだろうか。
まず考えたいことは、前述の器用貧乏の長所はいずれもヒューマンスキルであって、ビジネススキルやテクニカルスキルではないということである。
即ち、どのようなテクニカルスキルを持っていても、これらのヒューマンスキルがなければ、万能人、即ち我々が求めるフルスタックエンジニアにはなれないということである。
もっとも、フルスタックエンジニアになろうという人は既にこれらのスキルを持ち合わせていることが多いため大きな問題にはならないだろう。一方、これらのスキルを持ち合わせていない人にとっては、ヒューマンスキルを鍛えながらテクニカルスキルを磨くこととなるため険しい道となる。

百聞は一見に如かず

「百聞は一見に如かず」「論より証拠」という言葉や、似た言葉として「聞いて極楽見て地獄」という言葉がある。いずれも、論理や伝聞よりも体験が一番よく理解を促すという意味だ。
近年、インターネットではあらゆる情報が簡単に手に入るため、聞いただけで分かったような気になっている人も多いだろう。例えば、マイクロサービスという言葉もそうだ。「今後のシステム開発はマイクロサービスで作るべきだ」と言った論調の言葉を、記事やニュース、ブログ、講演会など多くの場で耳にする。しかし、こうした発言をする人の一帯何人がマイクロサービスを実体験として経験したことがあるのだろうか。
自分が経験したことのない内容をどれだけ語ろうとも、経験者の一言には敵わない。フルスタックエンジニアもそうだ。ただ知っている、聞いたことがあるという知識だけではフルスタックエンジニアとは言えない。幅広く豊富な知識があったとしても、経験に裏打ちされていなければ学者の域を出ないと言っても過言ではない。

つまりフルスタックエンジニアは、求められるテクニカルの範囲、または自分が求めるテクニカルの範囲について、一通りの経験があることが前提となる。

レベル４０とレベル６０の差

RPG(Roll Playing Game)をやったことのある人であれば、「主人公がレベル４０からレベル６０になるために必要なものは何か」と聞かれたら、即座に「経験値（Experience)」と答えるだろう。
多くのゲームでは経験値をためることでレベルアップし、次第に強くなって行く。もちろん、武器や装備といった経験値以外の要素もない訳ではないが、最終的には経験値が物をいう。

それではどうやったらこの経験値をあげる・ためることが出来るのか？多くの場合、経験値をためるには敵と戦う必要がある。ドラゴンクエストで言えば、最初はスライムなど弱い敵と戦いながら経験値を貯めレベルアップを図る。しかしレベルが上がるにつれて、弱い敵を相手にしているだけではなかなかレベルが上がらなくなる。そこで、そのレベルに応じた、さらに強い敵と戦い経験値を貯めていく必要がある。

エンジニアも同じだ。レベル４０の器用貧乏とレベル６０のフルスタックエンジニアとの決定的な違いは、その経験値にある。では、エンジニアのレベルはどうやって測ればいいのだろうか？
残念ながら現実世界においてこの経験値のレベルを定量的に測る手段は無い。何故ならば、どれだけ経

験値を積み、どれだけレベルアップしたかは結局のところ、勇者である本人にしか分からないからだ。「そんなことはない、当社には誰が見てもフルスタックエンジニアと呼べる立派なエンジニアがいる」と言う人もあるだろう。本人は謙遜してフルスタックエンジニアを名乗らないが、周りから見たら立派なフルスタックエンジニアは確かに存在する。

その通りなのだ。何故ならば、現実世界におけるレベルは、そのエンジニアが置かれた環境によって変わるからなのだ。例えば、レベル４０の勇者であっても、周りの敵がレベル１や２といった弱い敵しか存在しない場所では、間違いなく無双状態で活躍できる。ところが、物語も終盤のダンジョンの奥地に入ると、とても頼りなく周りの手助けなくてはすぐに死んでしまうキャラに成り下がる。

現実世界のフルスタックエンジニアも同様だ。自社の社員が全く開発できない部署にいれば、ちょっとプログラミングをかじっただけの駆け出しエンジニアであってもフルスタックエンジニアとして重宝される。そうした現場では誰もそのエンジニアを器用貧乏などと揶揄せず、皆がフルスタックエンジニアとして尊敬する。しかし、その彼もソーシャルゲーム開発現場のように日々納期との戦いに明け暮れるバリバリのエンジニアだらけの現場に入ると、「確かに色々知っているけど、その程度だと現場では使えないよね」等と言われ、簡単な仕事しか任せてもらえない器用貧乏になってしまう。
つまり、フルスタックエンジニアと呼ばれるためのレベルは、自身が置かれた環境によって大きく異なるのだ。

真のフルスタックエンジニア

ここで前述の「企業のためのフルスタックエンジニア」であれば、議論はこれで終わりである。
何故ならば、その企業のレベル感に合致した一定のスキルを身につければ良く、そのレベル感は各企業の IT リテラシーによって大きく異なるため、全国共通のスキルレベルの標準化はできないからだ。

しかし、その企業で働く「エンジニアのためのフルスタックエンジニア」を考えると、その先を考えなければならない。すなわち、どのような環境変化が起きたとしても、エンジニア個人としてフルスタックエンジニアと呼ばれる・又は自分自身で「私はフルスタックエンジニアである」と自負し続けるための努力が必要となる。

前述の理由から、フルスタックエンジニアとして活躍し続けるには、常に自分を高めるための環境が必要である。そのため、その環境を求めて、より高度で難易度の高いプロジェクトへ自ら進んで参加する方法もあれば、より上位を目指して転職するという方法もあるだろう。

何れにしても、常に自分の置かれた環境下でレベルアップするための努力をし続ける必要がある。
常に努力し、新しい物事に挑戦し続けることができるならば、そのエンジニアは、たとえレベルが低くともフルスタックエンジニアと呼ばれるだろう。一方、立ち止まってしまうと、その環境に合致したレベルにとどまる事となり、それまでフルスタックエンジニアと崇められていても、周囲の変化や環境の変化により、いつの間にか便利屋たる器用貧乏と呼ばれるようになってしまうだろう。

フルスタックエンジニアの定義

ここまでを整理するとフルスタックエンジニアの定義は次のようになる。

- 必要となるヒューマンスキル
  - イ） 頭の回転が早い
  - ロ） 要領がいい
  - ハ） 飲み込みが早い
  - ニ） 物事を習得したり覚えたりするのが早い
  - ホ） 好奇心旺盛
  - ヘ） 切り替えが早い
  - ト） 常に挑戦し続ける姿勢がある

- 前提条件

  以下の分野について一通りの経験がある
  - イ） OSI 参照モデルの第 3 層から第 7 層までの構築経験
  - ロ） 何かしらのアプリケーション開発手法論における一通りの開発手順の経験
  - ハ） 何かしらのプログラム言語によるシステム開発経験

- レベル感

  置かれた環境によって大きく異なる。

OSI基本参照モデル

7	アプリケーション層	第4層 アプリケーション層	HTTP	HTTPS	FTP	FTPS	POP3	SMTP	DHCP
6	プレゼンテーション層								
5	セッション層								
4	トランスポート層	第3層 トランスポート層	TCP / UDP						
3	ネットワーク層	第2層 インターネット層	IPv4 / IPv6						
2	データリンク層	第1層 ネットワーク インタフェース層	Ethernet				PPP	HDLC	ATM
1	物理層								

2. フルスタックエンジニアの育成

育成は出来ない

初めに宣言しておくと、フルスタックエンジニアの育成は出来ない。育成をしないと言っているわけではなく、育成出来ないと言っている。

フルスタックエンジニアに関わらず、厳密な意味でエンジニアの育成は出来ないと考える。我々にできるのはあくまで「場の提供」「機会の提供」であり、エンジニア自ら育つという意思がない限り育たないという意味だ。

これについては、経済産業省の IT 人材白書 2017 年版も次のように述べている。

> 企業が行わなければならないのは、誰もが挑戦できる環境、開かれた場を作ることである。
> そして個々の IT 人材は、自らも“デジタルトランスフォーメーション”の流れの中にあることを意識し、その中で活躍できる人材となれるように、自らの能力を高めることが重要である。そのためには情報への感度を高め、自ら挑戦する場を求める姿勢が重要になる。
>
> 出典：「IT 人材白書 2017　デジタル大変革時代、本番へ
> ～IT エンジニアが主体的に挑戦できる場を作れ～」
> 発行：独立行政法人情報処理推進機構（IPA）IT 人材育成本部

> 事業のデジタル化に必要な IT 能力を、既存の人材で賄うのは難しいとの意見があった。（中略）
> 新卒採用した人材を育成して人材確保する傾向が見られ、新卒を採用する際に理数系人材を重視する企業もいくつかあった。（中略）
> ただし、内部人材育成の難しさをあげる企業もあり、必要な技術を持った人材を中途採用できる場合は行い、出来ない場合はアウトソーシングや、外侮との連携を行うことで技術を補完する場合もあった。
>
> 出典：「IT 人材白書 2017　デジタル大変革時代、本番へ
> ～IT エンジニアが主体的に挑戦できる場を作れ～」
> 発行：独立行政法人情報処理推進機構（IPA）IT 人材育成本部

日本企業の場合、「就職」ではなく「就社」として会社に入るのであって、エンジニア個人が自分の意思で仕事を選ぶことができない現状もある。
しかもフルスタックエンジニアの特性として前章で述べた通り、基本特性としてヒューマンスキルが必須となる。このヒューマンスキルまで育成するとなると、現状では限りなく厳しいと言わざるを得ない。
そこでここでは「どうやって育成するか」ではなく「どうしたら育つ環境となるか」について論じたい。

従来の日本式育成

山本五十六の言葉に次のような言葉がある。

「やってみせ、言って聞かせて、させてみて、誉めてやらねば人は動かじ」

人材育成に関する名言として今でも良く耳にする言葉だ。これを文字通り実践している企業・人もいるだろう。ではこれを IT の現場、フルスタックエンジニアの現場に当てはめて考えるとどうなるか。
果たして、最新技術でかつフルスタックな領域について「やってみせれる」人材が何人いるだろうか？
自分では出来ないのに「やれみろ！なぜ出来ない？考えろ！もっと手を動かして！」などと言ってないだろうか。

「俺らが若い頃は・・・」と言う人もいるかもしれないが、いまは全く時代が違うとしか言いようがない。現代の IT は過去類を見ないほど早い速度で変化している。アプリケーション開発言語のみならず、開発環境、開発手法論など変化が激しい。ムーアの法則ですら通用するのは 2021 年が限界だ(注 2)とまで言われている。その上、ビジネスの変化が早いため求められる納期も短く、以前のようにウォータフォールでじっくり時間をかけた開発というわけにも行かなくなってきている。このような時代のエンジニアに対して、ベテラン社員の「昔取った杵柄」を持ち出す話は、全く役に立たないどころか、「年寄りの昔話」として相手にすらされないだろう。
つまり、これまでのような「根性と気合」で乗り切る方法や、講師が一方的に教育する方法、「俺を見て育て・技術は盗め」的な方法では現代 IT 分野におけるフルスタックエンジニアを育てることはできないのだ。これが前述 IT 人材白書にもあった「内部人材育成の難しさ」であると考える。

教育ではなく学習

ところが、この山本五十六の言葉には続きがある。

「話し合い、耳を傾け、承認し、任せてやらねば、人は育たず。」
「やっている姿を感謝で見守って、信頼せねば、人は実らず」

やはり山本五十六は凄いと恐れ入る。人を一人前にするには、任せてやらせなさい、そしてその姿を見守りなさいと言っているのである。つまり山本五十六は「部下を教育するのではなく、部下の学習を助けよ」と言っているのだ。

育成とは「教育」ではなく、そのエンジニアが自ら「学習」するための場を提供し、機会を与え、気づきを与えることだと筆者は考える。

「出来た」の手前にはたくさんの「出来ない！」「わからない！」が必要

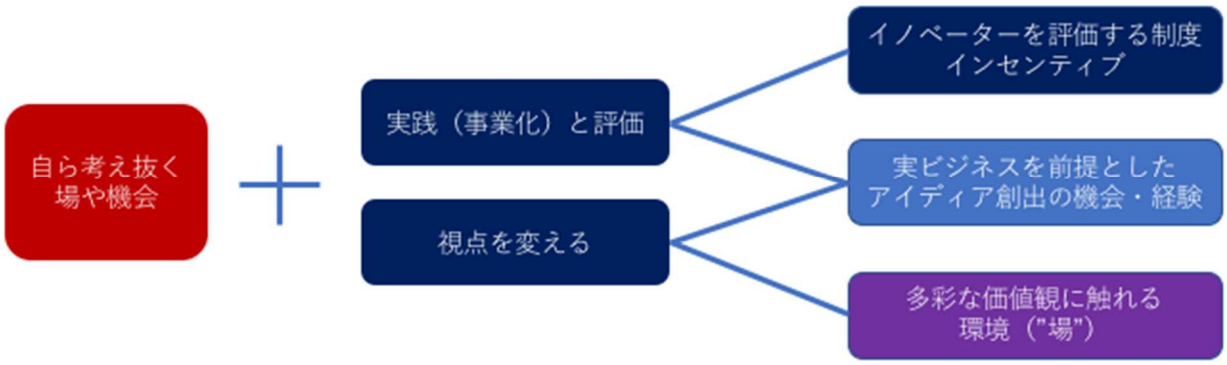
これは子供達に「アハ体験」(注3) を如何に与えるかに注力している「宮本算数塾」の塾長宮本哲也氏の言葉である。さらに宮本氏はこう続ける。「算数は解けることを楽しむ科目ではなく，解けないことを楽しむ科目です。」と。つまり宮本氏は、「簡単に答えを出すことよりも、自らの力で考え、間違えを繰り返しながら自らの力で答えにたどり着くことの方が大切だ」と言っているのである。この言葉は、小学生向け算数塾での話に限ったことではない。

職場でも新入社員に対して「失敗させない」「苦労させない」「先に答えを教える」「先に手助けする」という現場を多々目にする。明らかに新入社員が「学習」する機会を先輩・上司が奪っているに他ならない。
先輩・上司とすれば「指導している」「教育している」つもりになっているのであろう。しかし、一方の新人からしてみれば「大きなお世話」なのだ。

「仕事で失敗されたら困る」という言い分もわかる。しかし、育てたいと思うのであれば、失敗することが分かっていても敢えて見守る覚悟が必要なのだ。そして、これこそが山本五十六の言葉の本質ではないだろうか。

経済産業省がまとめた次世代高度 IT 人材に関する育成フレームによると、次世代高度 IT 人材の効果的

な育成手法に求められる要件として、「自ら考え抜くための場や機会」の存在がまず前提となるとある。



フルスタックエンジニアについても同様である。場を提供し、実践する機会を与え、その結果を評価する制度・インセンティブを与える必要がある。また、多様な価値観に触れるための環境の提供も必要だ。

具体的には、自らの力でアプリケーション開発に関する一通りの手順を踏む場を提供するのだ。ここで、インフラだとかアプリといった区分けは不要である。そのような区分けはこれまでの分業化時代の名残であり、これからフルスタックエンジニアを目指す若手にとっては無用な制限となる。
これまでの規定・規制・価値観・区分けに囚われることなく、ただゴール設定をしてあげれば良いのだ。もし、ここで「私は習ってないので出来ません」「それについては教えてもらっていません」などというエンジニアが居たとしたら、残念ながら彼らのフルスタックエンジニアへの道は遠いだろう。なぜならば、彼らには先に挙げたヒューマンスキルが備わっていないからだ。

会社としてフルスタックエンジニアを育成したい、確保したいと考えるのであれば、場の提供だけでは不十分だ。それに応えるための制度やインセンティブが必須であるし、彼らが成長し続けるための仕組みも考える必要がある。

一方、エンジニア個人として考えれば、１００％完璧な環境などあり得ないと心得、自分の置かれた環境で、いかにフルスタックエンジニアとして成長するかを考えながらスキルアップを目指せば良い。
その上で、自分が育たない・評価してもらえない環境であれば転職すれば良いのである。例え勇者としてのレベルが低くとも、置かれた環境で自他ともに認めるフルスタックエンジニアになっていれば、他の環境で評価される機会は十分にある。

「フルスタックエンジニア」の育成とは

今回のテーマは「新卒 3 年でフルスタックエンジニアを育てる」である。この入社して 3 年という期間

を短いと考えるか、それとも長いと考えるかは、結局のところエンジニア本人がどう考えるか次第だ。現在の IT の世界で 3 年という期間は長い。3 年前の技術が使えなくなっていたり、3 年前には想像もつかなかったような技術が簡単に手に入ったりすることもある。

また、システム利用者であるエンドユーザーの IT リテラシーも年々高くなっている。年齢層に関係なく誰しも普通にスマートフォンを使いこなし、自宅に帰ると AI 搭載のスピーカーが出迎え、ロボットの掃除機が自宅を掃除する時代なのだ。当然ながら各システムに求められるニーズも高くなっている。
そのような時代変化の中にあって「入社して 3 年間は育成期間なので実務では使えません」などという言い訳が通用しない世界になっている。

企業としては新入社員を 3 年程度で一人前にしたいという想いがある。一方、彼ら全員がフルスタックエンジニアにはなれないということも重々わかっているし、入社試験でそれらの素質を 100%の確率では見抜けないことも分かっているはずだ。しかし、そうだとしても企業は継続的にフルスタックエンジニアの育成に投資していかなければならないのだ。

企業が一人でも多くフルスタックエンジニアを排出するには、彼らが育つ制度・環境、あるいは企業文化を整備する。その上で、フルスタックエンジニアたるヒューマンスキルを備えたエンジニアを配置するしか方法がない。その上で、彼らが常にレベルアップし続けられるように、仕事を任せ、信頼し、暖かく見守ることこそが、フルスタックエンジニアを増やすための一番の方法である。

一方、個人としてフルスタックエンジニアを目指すのであれば、先のヒューマンスキルの研鑽に心がけ、色々なことに挑戦し続けるしかない。この挑戦する場が与えられなければ、別の場を探し、そこへ移ることが最も効果的にレベルアップする方法だと筆者は考える。

以上

注 1 ：GGRKS
ggrks. 「インターネット検索しなさい」の意。 最大手インターネット検索エンジン Google(グーグル)の普及によって、ネット検索すること自体を本家グーグルから離れて「ググる」というが、電子掲示板やチャット、SNS などで、なんでも聞きたがる人に対して「ググれカス(野郎)」と使われる。
出典：コトバンク（https://kotobank.jp/word/ggrks-188462）

注 2 ：ムーアの法則は 2021 年が限界
出典：「国際半導体技術ロードマップ」（ITRS）

注３：アハ体験（心理学）

アハ体験（ドイツ語: Aha-Erlebnis）とは、ドイツの心理学者カール・ビューラーが提唱した心理学上の概念で、未知の物事に関する知覚関係を瞬間的に認識する事を指している。洞察を研究する心理学では、認知過程が完了した後に現れる特徴として、その体験がしばしば現れるとされる。

出典：Wikipedia (https://ja.wikipedia.org/wiki/%E3%82%A2%E3%83%8F%E4%BD%93%E9%A8%93_(%E5%BF%83%E7%90%86%E5%AD%A6))

以上

2018 年 3 月 31 日

JUAS アドバンスド研究会

「新卒３年でフルスタックエンジニアを育てる」

要旨

スマフォに代表される情報端末が個人の生活に浸透密着し、カメラ・マイク・その他多くのセンサーが通信と融合した IOT デバイスがあふれる現在、もはや IT は、単なる既存ビジネスプロセスの自動化や省力化のためのツール（人間の仕事を代替するツール）ではなく、それ自体がビジネスプロセスの発信源であり、ひいてはビジネスモデルすら主導する存在になっている。

こうした変化は、IT 人材に求められるタレント（才能や性質）やスキル（能力）にも大きく影響を与えるはずだが、国内 IT 業界（ユーザー系、ベンダー系問わず）の人材育成では相変わらずスペシャリスト志向が強く、OFFJT では OS や RDMS や最近ではクラウドなどの利用方法、あるいはプロジェクトマネジメント手法などを学ばせることが中心であり、OJT では自分が担当する（既存）システムの機能や構造などを経験的（場当たり的）に学ぶといったことが続いている。この状況が続く原因は「人から言われたものを作る」受託開発では、こうした人材育成方法が今も非常に有効だからである。

しかしこのスタイルは結局「欧米 IT 製品の優秀な使い手」を育てているだけで、現在求められている「ビジネスプロセスやビジネスモデルの変革を主導できる人材」の育成にはつながっていないのではないだろうか。

実際、日本発と胸をはれるメジャーなイノベーションは近年あまり登場しておらず、ただ欧米の後塵を拝し、欧米発のサービスや製品の勉強や利用に高い金を払う状況が続いているように見える。

この状態が続けば、IT 業界のみならず、日本全体が IT イノベーションを主導できない国、他国が開発した IT イノベーションを高い金を払って使わせてもらうだけの国に成り下がるかもしれない。いや、すでになりつつあるのではないか。

（日本が負け続けるのは、もうコリゴリだ。２０世紀の成功体験は捨てて今こそ新しい戦略をもつべきだ。本邦 IT 業界の状態はあまりにもひどい、幕末の幕府のようだ）

本稿は、こうした危機感を背景に、まず現在の IT 人材の最重要課題を「開発スピードの超高速化（アジャイル)」と「ビジネスイノベーションの実現」への対応と前提し、それに対応できる人材の基礎的・共通的なイメージとして「フルスタック・エンジニア」を定義し、その育成方法と育成課題について、英国の小学校における戦略的な取組みや、マイコン黎明期における本邦での技術教育にも触れつつ論じる。

＊イノベーションのジレンマ

習熟や上達に対する志向が非常に強い我々日本人は、性能改善や機能改善などの、インクリメンタルイノベーションに拘泥する傾向が強く、ラディカルイノベーションの足音が近づいていることに気づきにくい（拡充・改善も確かにイノベーションには違いないが、今求められているのは、それではない。ここで論じようとするものも、もちろん違う）

あんなものはまだまだ使えない、そう思っていたテクノロジーが急速に安価で高性能になっているのに、いつまで従来の技術や方法を捨てられない。

いま一番大事なことはもしかすると捨てることかもしれない。(使い捨てシステム化。いつまで、１０年以上使うつもりでシステム開発をやるのか)

１．IT 人材の主要課題と対応

「IT 主体でビジネスプロセスやビジネスモデルを変革する」ために取り組むべき課題。それを挙げたらきりがないかもしれない。しかしその中で特に緊急性が高い課題を選ぶとしたらほとんどの IT 関係者は「開発スピードの超高速化」と「ビジネスイノベーション実現」を挙げるのではないだろうか。(JUAS の IT 動向調査などから)

日本は両方の点で残念ながら欧米や中国に遅れつつある。そのことはもう認めなければならないだろう。

（１）開発スピードの超高速化

・なぜ必要か？

ビジネスシステムであれ公共のシステムであれ、それを開発あるいは導入する目的は、何かしらのメリット（経済効果、宣伝効果、など）を享受することである。したがってシステムは一分一秒でも早く出来上がった方がよい。その分得られるメリット（特に経済効果）の総量が増えるからだ。そのためシステムの受託開発契約では「納期遵守」が実は最大の「契約債務」であり、本来、いかなる手段を講じてでも達成すべき課題である。

ところが、「納期遅延」が頻発する我が国では、これが意外と忘れられている。それは即損害賠償につながるということを IBM vs スルガ銀行裁判の結果などを踏まえて、我々はあらためて心に刻み込むべきだろう。

・何が必要か

では、開発スピードの高速化を実現するために、最も効果的なことはなんだろう。

それはずばり開発チーム内のコミュニケーション（情報共有など）にかかる時間を減らすことである。システム開発における最大のタイムイーターは、コミュニケーション・コストだからである。

コミュニケーション・コストを減らすといっても単純に会議時間を減らせばよいということではない。それでは必要な情報が共有されず、システム開発はうまく進まない。

一番効果的な方法は、結局、開発チームを小さくすることだ。１００人の開発体制と、３人の開発体

制を比べたら、どちらの方が情報共有のための工数の割合が増えるか考えれば自明だろう。ソフトウエア見積手法 COCOMO の計算式でも、開発チームの規模が小さいほど開発生産性が高い（コミュニケーションのためのロスが少ない）としている。

では開発チームを小さくする（マイクロ・チーム化）にはどうしたらよいか。

それには二つのアプローチがある。

ひとつは単純に開発規模（プロジェクトスコープ）を小さくすること（マイクロスコープ化）だろう。アジャイルとかマイクロサービスなどという最近流行りのバズワードも、少々視点が違うものの、突き詰めれば同じことをいっている（マイクロスコープとマイクロ・チームの組み合わせだから迅速な開発も可能になる）。

二つめはマルチタレント・マルチスキルの要員を集めることで、チームの必要人数を減らすことである。ひとり一役しかできない要員だけでチームを組むのと、ひとりで何役かこなせる要員を集めるのとでは、必要な人数が違うのは当たり前だ。そして、その究極は「ひとりで全部（ハードも OS もネットワークもビジネスロジックも UI も）の仕事をやれるエンジニア」にきてもらうことである。

米国の GAFA（Google, Amazon, Facebook, Apple）はいうに及ばず、アジアの HUAWEI などでも、能力があれば新卒の社員にもポンと年間１０００万円以上の給料を出すと言う。彼らが求める人材も「ひとりで全部できる人材」であろう。一方、日本では、未だにスペシャリスト志向が主流に見える。大卒初任給は、２００万円台程度だろうか。国際競争に勝てる気がまったくしない。

HUAWEI などが１０００万円出す「ひとりで全部できるエンジニア」は、いわば＜天才的＞「フルスタック・エンジニア」であり、本稿で育成を提案する＜入門的＞「フルスタック・エンジニア」とは異なる。

天才はあくまで、才能と個人の努力によって誕生するものであって、企業内の教育で人工的に育成するのは難しいだろう。それは、この研究プロジェクトの中で何度も話題になった。

しかし企業でもなんとか育てられそうな＜入門的＞「フルスタック・エンジニア」であっても無駄にはならない。開発チームの生産性を高めるのには十分効果がある。そして、確率はたいへん少ないかもしれないがその中から＜天才的＞「フルスタック・エンジニア」に進化する人材が現れる可能性もあるだろう。

だから、その育成を提案するのである。

（2）ビジネスイノベーションの実現

・なぜ必要か？

JUAS の IT 動向調査でも、IT および IT 部門に対する要求で一番多いのがイノベーションの実現である。CIO とはもともと、Chief Information Officer（最高情報責任者）
の略であろうが、最近は Chief Innovation Officer（最高イノベーション責任者）の略と言われることもあるようで、いかに IT が期待されているか（イノベーションを丸投げされている？）よくわかる。

では期待されているイノベーションとはどういうものか？

それが分からないからみんな苦労している。

過去のイノベーションを分類して、プロセスイノベーションだ、プロダクトイノベーションだ、インクリメンタル（改善型）だ、ラディカルだ（革新型）など様々に言うことはできるが、これから発案しようというものを事前に説明できるはずがない。イノベーションとは、いつの時代も事後的に認定されるものである。

ただ、ヒントはありそうだ。

たとえば目的から考えるということ。イノベーションの目的（あるいはそれがイノベーションと認められる理由は）は結局のところ「競争力の源泉」になるかどうかである。

つまり単純な自動化であってもそれが他人にはできないことならイノベーションを実現したと言えるが、逆に AI や IOT などを駆使した複雑なシステムであっても、クラウドサービスとしてすでに安価に提供されていて誰でもが簡単に使えるようなものなら、そこに使われている技術が最新であろうとすごかろうと、もはやそれを導入したからといってイノベーションを実現したとはいえない。競争優位につながらないからだ。

＊ Google などのクラウド大手は、運用の自動化「NO OPS」を謳い文句にしはじめているが、その先にあるものは間違いなく開発も運用も自動化される「NO DEV NO OPS」であろう。知らず知らずのうちに IT 帝国に引き込まれている。

日本では未だに、最新テクノロジーさえ導入すれば、イノベーションを実現した気になる人が多いようだが、まったくの勘違いと言わざるを得ない。それこそ日本がイノベーションで立ち遅れている原因のひとつであろう。

「競争力の源泉」足るには、結局「独創的」でなければならない。

・何が必要か？

では上記のイノベーションを実現するにはどうしたらよいか。

身も蓋もない話だが、最終的には天才的なアイデアの出現を待つしかない（日本の場合は、天才の足を引っ張らないことの方が大事かもしれないが）。凡人が日頃から思いつくようなことでは、競争力にはならないからだ。

前にも書いたが、（企業の）教育で天才的な着想を得る人材を育てることはかなり難しいだろう。

教育とはしょせん土台を作る地道な作業で、それを受けたら、いきなりひらめくというものではない。

ところがIT業界ではそういう認識が乏しいようで、すぐに効果を求めてしまう。結果、教育の中心はスキル系（スペシャリスト系）になり、とりわけ人気なのはベンダー資格のためのセミナーになる。なんとかプラチナ資格を取った人材は「ある製品の使い方をよく理解している人」であり、もちろん現在のIT現場では重宝される人材ではあろうが、イノベーション実現のための準備が整った人材とはいえない。にもかかわらずそういう人材に、イノベーションをやれやれ言っているのが日本の現状ではないだろうか。

＊士官学校を卒業した軍人にいきなり政治をやらせるようなもの。

ではイノベーションにつながる教育とはどんなものだろうか。

ここであらためて「独創的なアイデア」とは何かを考えてみよう。

それは世の中に存在しない「新しい組み合わせを発見」することではないだろうか。

筆者は新入社員のときそれを「異質の統合」と習った。

その実例として教えてもらったのがミシュランの話である。グルメガイドで有名なミシュランは世界初のタイヤメーカーである。その創業者は「ゴム風船」と「車輪」を組み合わせて、タイヤの原型を発明した。そんな組み合わせは誰も思いつかなかった。開発は困難を極めたが、改良を加えて完成した。車のタイヤに代わるものは未だに発明されていない。

ここから得られる教訓はふたつあると思う。

ひとつは「独創的なアイデア」生むには「異質な知識（情報）」をたくさん頭に入れておいた方が有利ということである。

日本では、古来、究道的な姿勢や人物を尊ぶ傾向があり、何かひとつ極めれば、あらゆることに対応できる（「一道に通ずれば、すなわち十道に通じる」）といわれるが、こと「独創的なアイデアを思いつく」に関しては、浅くとも広い知識をもつ方が有利ではないだろうか。

ITエンジニアなら、ハードからアプリまで一通りは知っている。語れる。ついでに、経済や経営の知識があればなおよいのではないか。

二つ目の教訓は、組み合わせにタブーはない、いやむしろタブーにこそ「独創性」が宿るということである。

つまり先入観でタブーを排除しない、タブーにあえて挑戦するという習慣も基礎教養として身につけるべきことになるだろう。

２．入門的フルスタック・エンジニアとは？　まとめ

前項で「入門的フルスタック・エンジニア」の概要について述べたが、ここでその主要属性（到達目標）を整理しておこう。

（１）知識面

・ハードからアプリまで、システムのことはいちおう全部知っている。
・　ひとりでもシステム開発ができる。
・　経営や経済などの関連知識も、ひととおりある。
・　発想法や、文章作成法などの知識もある。
・　知らないこと、わからないことを学ぶ方法を知っている。
・　発明の方法を知っている。

（２）性格・精神面

・想像力が豊か。
・　好奇心旺盛。
・　失敗してもめげない（イノベーションの成否はアイデア数が勝負）。
・　先入観や思い込みが少ない。

３．教育方法について

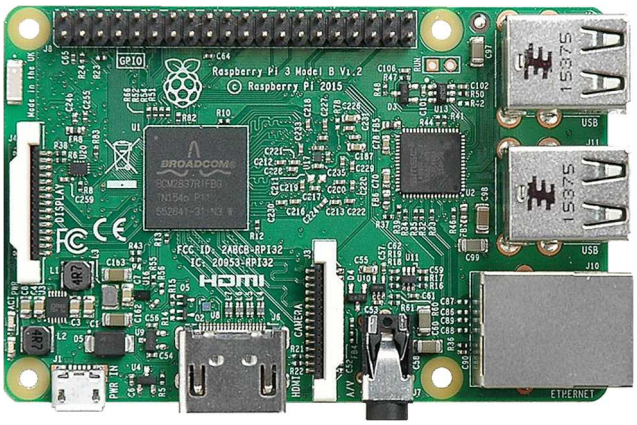
教育方法については、他のメンバも詳しく書いているので、ここではハード教育についてのみ書く。

「入門的フルスタック・エンジニア」の技術教育では、ハードからソフトまで一通り習得させることが目標になる。ところが日本の IT 技術者教育はソフトが中心で、ハードについては PC やルーターの設定ぐらいしか教えていないのが実状である。

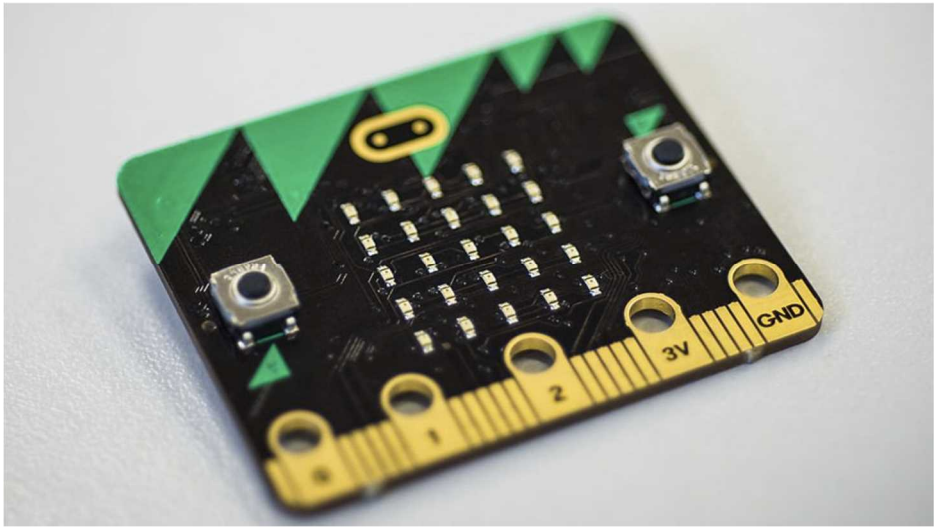
この状況は、日本では学校でのコンピュータ教育も「プログラム中心主義」であることと軌を一にするのではないかと思う。

＊下調べの段階からプログラム教育中心。「諸外国における プログラミング教育に関する調査研究」 （文部科学省平成 26 年度・情報教育指導力向上支援事業）

これに対して特に英国では、３０年以上前から、ハードとソフトのハイブリッド型の教育を行っている。そこから生まれたものが、日本でもよく知られている Raspberry Pi（ラズベリーパイ）である。



そして２０１５年には、BBC（英国放送協会）が「Make it digital」と称して、Micro:bit というボードコンピュータを英国に住むすべての 11 から 12 の子供たちに無料で配布している。Micro:bit はその後 Microsoft に移管され、維持管理を行っており、現在は日本でも入手可能である。２～３千円で購入できるものながら、モーターやセンサーなども接続でき想像以上に応用範囲は広い。



http://www.bbc.co.uk/programmes/articles/4hVG2Br1W1LKCmw8nSm9WnQ/the-bbc-micro-bit

こういうもので遊び倒して力をつけた連中と日本の現在の小・中学性は将来渡り合わなければならない。不安になる。

日本の初等教育でも、お絵かきソフトのようなものや、スクリプト言語ツールみたいなものだけではなく、こういうものを早急に取り入れて、ハード操作の力をつけなければならないと思う。

ちなみに ARUDUINO はイタリア発祥、Mind Storm はデンマーク製である。コンピュータはアメリカという印象が強いが、実はヨーロッパもこうして着々と次世代の教育を進めているのである。奥深さと戦略性を感じるとともに、またしても日本の遅れを感じてしまう。この種のものが日本にないのはまさにハード軽視の表れでもある。なんとか改善したい。

IT 企業での技術教育でも、今のところ上記で紹介したものよりよいものは見当たらないため、実は、筆者の勤務先でも Mind storm や Raspberry PI を使った研修などを行っている。

どんな成果につながるか。これからが楽しみである。

以上

2018 年 3 月 31 日

JUAS アドバンスド研究会

「新卒 3 年でフルスタックエンジニアを育てる」

1. フルスタックエンジニアが必要になった背景

(1) 市場の変化

「破壊的テクノロジが、事業のファンダメンタルを変える可能性がある。それがオープンな形で普及すれば予測できない影響がでる」(平井 一夫, 社長兼 CEO, ソニー株式会社, 日本)

いま、世界のトップ経営者が脅威と感じていることの一つが 「ウーバー症候群」である。ウーバー (Uber) は、乱暴に言えばスマートフォンを使った白タクシステムであり、IT による規制企業サービスの中抜きであり、コンシューマ (乗客) とリソース (自動車所有者) を、ダイレクトに、低コストで、安全に結びつけた事例であり、コンシューマ、リソース、サービス運営者 (Uber) の三方良しを実現した破壊的なサービスである。

これは、ビジネスモデルがまったく異なる競合が市場に参入し、既存企業を破壊していく現象である。競争の境界線があいまいになりつつあり、競合はどこにいるかわからない世界になりつつある。そうなった時には、新たなプレーヤー、新たなビジネスモデルが既存企業に取って代わる。

(IBM グローバル経営スタディー レポート 「境界線の再定義」より
http://www-935.ibm.com/services/jp/ja/c-suite/)

例えば、エネルギー寡占からの「破壊的なシフト」も起こり得ないわけではない。そうならないためには、各企業が既存のビジネスに固執する経営マインドを転換し、強みを有する領域に経営資源を集中し、自前主義を脱して他の領域で強みを有する他者を巻き込み、付加価値の高いエコシステムをいかに作り上げていけるかが鍵となってくる。

このようなビジネス変化の中、B2C ビジネスにおけるデジタル (IT) の役割は確実に増大している。既存のビジネスモデルにおけるデジタルの役割はもちろん、IT 自体が主役となって新たなビジネスモデルが急速に発展している。このようなビジネスのトレンドをデジタルビジネスと呼ぶ。

デジタルビジネスを支える主要なテクノロジは、ソーシャル、モバイル、IoT (モノのインターネット)、ビッグデータとアナリティクス、クラウドの五つである。

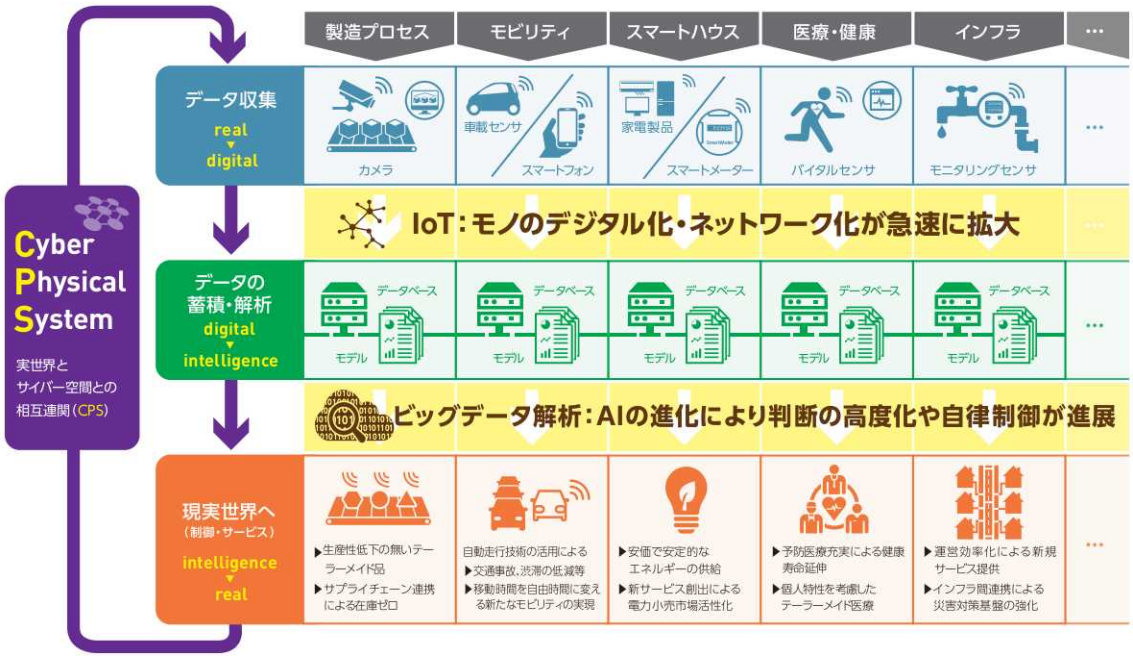
① ソーシャル

ソーシャルは完全にマスマーケティングの主戦場の一つになった。ブランドは、Facebook、Twitter、Instagram などのソーシャルメディアを通して、より適切なユーザに、より適切な情報を、より安価に配信することが可能となった。また、大衆やユーザの声を、よりダイレクトに、より大量に収集しることが可能となり、これらの情報を分析することで、正確なマーケティング効果の測定、適切な商品・サービスの改良、新たなビジネスモデルの発想など、様々な活用が進んでいる （ビッグデータとアナリティクスとの連携)。

② IoT （モノのインターネット）

スマートメーター、モバイルデバイス、ウェアラブルデバイス、様々なセンサーからネットワークを通して情報を収集する。その情報を様々なアルゴリズムで分析し、機械が自動であるいは人の判断で、ネットワークを通してデバイスを制御することがきる。このようなことが、低コストでできる世界が間近に迫っている。ここで言うデバイスとは、スマートメーター、HEMS 機器、自動車および車載センサー、スマートウォッチ （人体の基礎バイタル情報の収集)、監視カメラ、生産機器、産業機械などで、これらの情報を活用した様々なサービスやビジネスモデルが発展しつつある。

このように実世界をデジタルデータに変換し、そのデータを処理した上で、現実にフィードバックするというループを実現することを Cyber Physical System （CPS） と呼ぶ。



図：Cyber Physical System によるデータ駆動型社会の概念図
（出典：経済産業省 産業構造審議会 「CPS によるデータ駆動型社会の到来を見据えた改革」）

③ モバイル

カスタマージャーニーのライフサイクルを広くカバーするキーデバイス。モバイルアプリで高度な UX （ユーザエクスペリエンス、顧客経験） を提供し、顧客の行動や嗜好をきめ細かく把握する。カメラ、GPS、各種センサーなどのデバイス機能を活用し、コンシューマとプロバイダーをダイレクトに結びつけるチャネルとして B2C ビジネスの最前線に位置する。

④ ビッグデータとアナリティクス

スマートメーター、モバイルデバイス、ウェアラブル デバイス、SNS などから収集される大量のデータを、捨てることなく全て蓄積し、全て分析することが可能な時代が到来した。これにより、従来は統計的に見過ごされていたロングテール領域へのアプローチや、きめの細かい One-to-One マーケティングが可能となった。大量データを機械学習アルゴリズムに処理させることで、深層機械学習 （Deep Learning） の急速な進化が進み、従来人間が行っていた判断業務 （ローン審査、投資判断、医療診断、画像解析など） が機械にとって替わられる時代が一部現実になりつつある （融資の可否を 5 分で自動判断する FinTech 企業、世界チャンピオンに勝利した囲碁ソフト AlphaGo など)。

⑤ クラウド

クラウドを活用（= コンピュータ資源を従量課金でアウトソーシング）することで、探査・新規ビジネス領域へのコスト的、技術的参入障壁を大幅に下げることが可能となった。クラウドベンダーは、IaaS （ハードと OS のインフラのみ)、PaaS （ミドルウェアまで)、SaaS （アプリケーションまで） など様々なレベルのサービスを提供しており、ユーザはこれらのサービスを組み合わせることで、短期間で新たな IT 環境を調達、構築することができる。また、クラウドベンダーが次々にリリースするサービスを利用することで新たなテクノロジへの追従も容易となる。これは、IT がこれまで以上にキーパーツとなるこれからの B2C ビジネスにおいて非常に重要なことである。

またアプリケーションエンジニアにおいてもクラウドの普及によって、これまでの開発スタイルが大きく変化することとなった。

(2) クラウドの普及

近年クラウドサービスの普及により、アプリケーションエンジニアがサーバ構成を決めたり、チューニングを行ったりすることができる。

AWS（Amazon Web Services）や Microsoft Azure などのクラウドサービスの登場で、インフラの構築はハードウエアの話ではなく、ソフトウエアの話に切り替わっている。サーバーサイドでは、Node.js など、フロントエンドでは AngularJS といったフレームワークや、SASS、CoffeeScript、Haml などの可読性の高いプログラミング言語の登場。さらに、デザイン面では Bootstrap などのフリーのツール集も普及している。Web サービスを作るために必要な技術がコモデティ化し、学習コストは以前に比べると劇的に下がっている。

このようにインフラの構築からアプリケーションの設計、実装、デザイン、デプロイ、運用というシステム開発における一連の流れは、1 人で十分こなせる範囲になってきている。

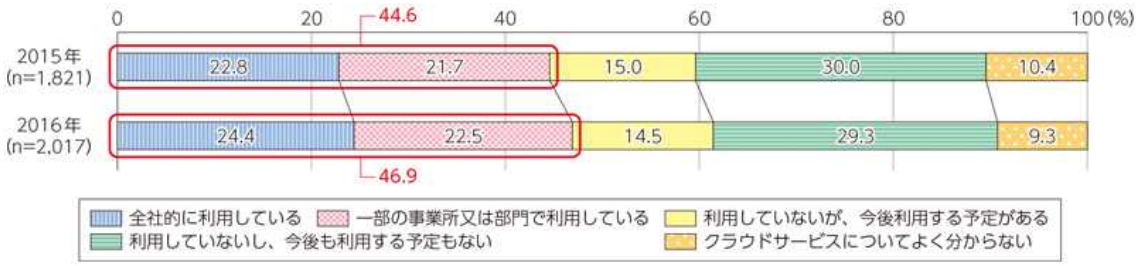


図　企業におけるクラウドサービスの利用状況

（出典）総務省「通信利用動向調査」
(http://www.soumu.go.jp/johotsusintokei/statistics/statistics05.html)

しかし、上記のように技術革新が日進月歩で進んでいるものの、実際に作業するエンジニアが不足している。

(3) IT 人材の不足

我が国では、産業界で大型の IT 関連投資が続くことや、昨今の情報セキュリティ等に対するニーズの増大を契機に、IT 人材の不足が改めて課題となっている。また、ビッグデータ、IoT 等の新しい技術やサービスの登場により、今後ますます IT 利活用の高度化・多様化が進展することが予想され、中長期的にも IT に対する需要は引き続き増加する可能性が高いと見込まれる。

しかし、我が国の労働人口（特に若年人口）は減少が見込まれており、今後、IT 人材の獲得は現在以上に難しくなると考えられます。IT 需要が拡大する一方で、国内の人材供給力が低下し、IT 人材不足は今後より一層深刻化する可能性がある。

IT の我が国産業の成長にとっての重要性を踏まえると、今後も十分な IT 人材を確保することは、我が国にとってきわめて重要な課題であるといえる。

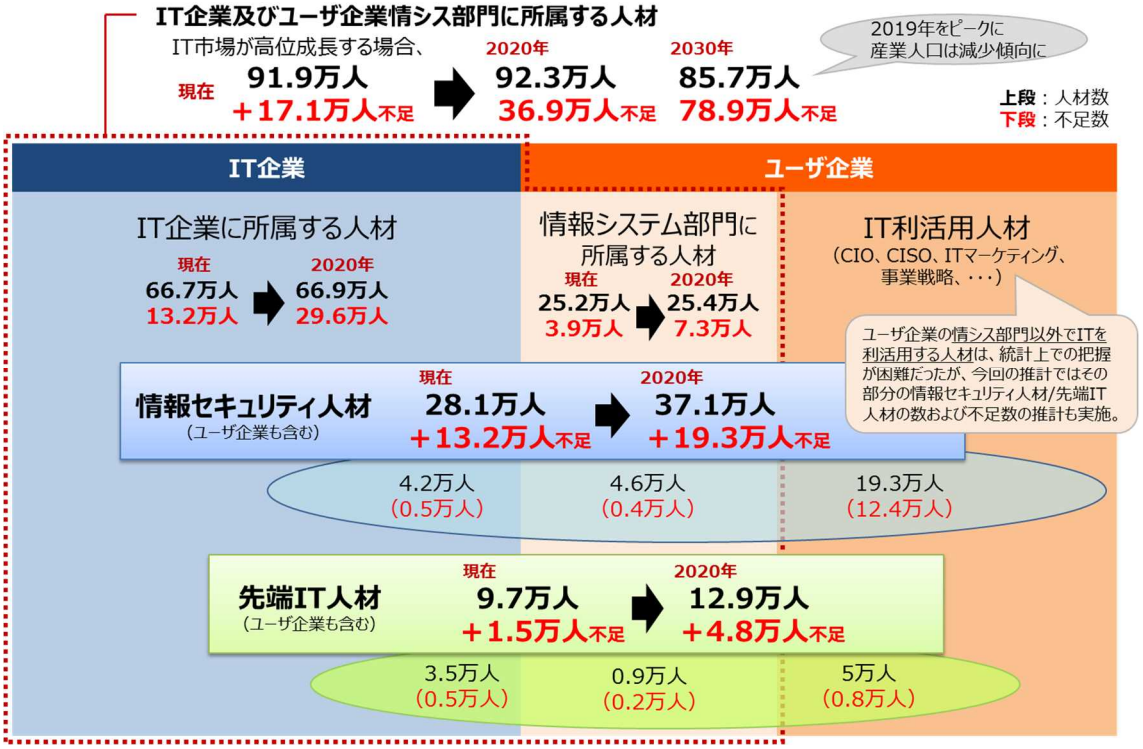


図　IT 人材の需給に関する推計結果の概要

（出典）経済産業省「27 年度調査研究レポート」
(http://www.meti.go.jp/policy/it_policy/jinzai/27FY_report.html)

(4) エンジニアに求めるスキル

前述のような、市場の変化、技術の変化、慢性的な IT 人材不足という状況から、現在、そして今後特にエンジニアへ求めるスキルは、0 から新たに新規事業を支えるシステムを柔軟・迅速に開発できることである。既存事業を支えるシステムの機能拡充や運用と異なり、新規事業を支える「デジタル化」はトライアンドエラーで要求を追求し、都度改変していかなければならない。

従来型（ウォーターフォール）での開発プロセスやこの開発モデルに慣れたエンジニアだけでは、市場を破壊するような価値を生み出すことは困難であると考える。

そのため、「デジタル化」の世界においては、アーキテクチャの選定からインフラ構築、設計、開発、運用を短いサイクルで実現するアジャイルなシステム開発が求められる。

このように、既存事業を支えるシステムと新規事業を支えるシステムでは、特性が大きく異なることから、ガートナーでは、要件に応じてシステムの特性を分けて考える「二つの流儀」という考え方を提唱している。「モード 1」は変化が少なく、確実性、正確性、高品質が求められる領域のシステム。ある程度、コストが高くなったり、サービスの価格が高くなったりすることを許容してでも、確実性や品質を重視するシステムである。象徴的な例としてはメインフレームなどが挙げられる。「モード 2」は、品質はそこそこで良いとし、その分、開発・改善のスピードや、より低いコスト、価格競争力、何かあったらサポートなどがすぐに対応する「使いやすさ」などを重視するシステムである。

モード 1、2 ともにユーザの「満足度の高さ」を狙うことは同じだが、モード 1 が「安心・安全」を狙うのに対し、モード 2 は「すぐに分かる、できる、使っていて楽しい」といった要素を優先する。

モード 1 のシステムを開発するエンジニアとモード 2 のシステムを開発するエンジニアでは、開発手法も違えば、必要となるスキルセットも異なる。

現在各企業が求めているエンジニアは、「デジタル化社会」を推進するエンジニアであり、このエンジニアこそフルスタックエンジニアであると考える。

2. フルスタックエンジニアの定義

「クラウドで提供するサービスを駆使してエンタープライズアプリケーションを構築できるエンジニア」

クラウド時代においては、インフラエンジニア、ネットワークエンジニア、アプリケーションエンジニアいずれにおいても、ネットワークからアプリケーションまですべてのレイヤーを体系的に理解する必要がある。

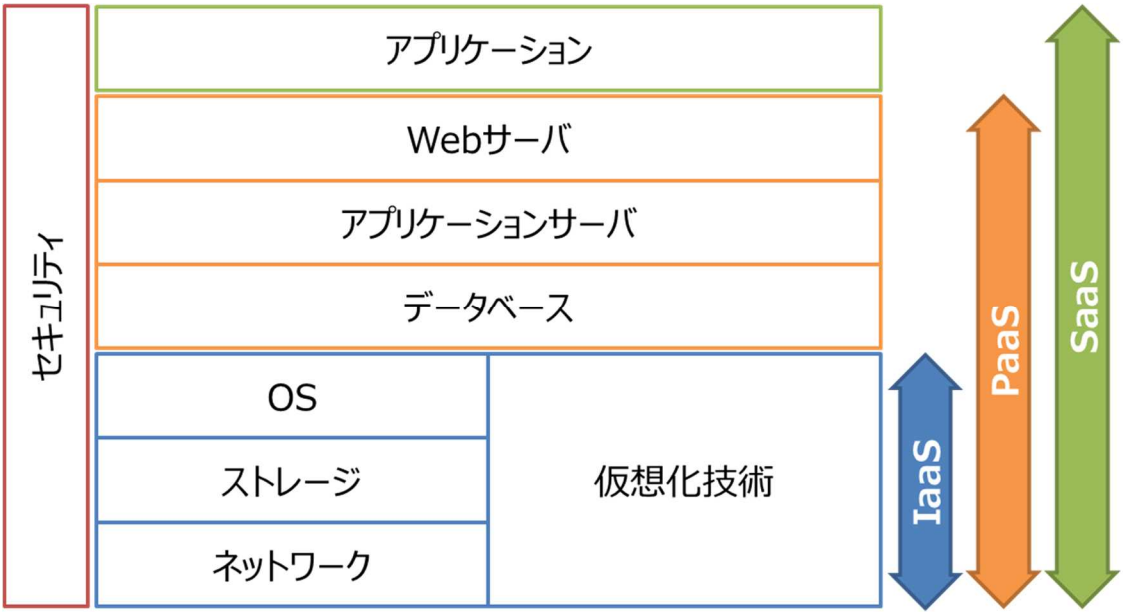


図　クラウドサービスがカバーする範囲

このようにクラウド時代のエンジニアに求めるスキルセットが拡大していることから、これらすべてを体系的に理解し、クラウドで提供するサービスを駆使してアプリケーションを構築できるエンジニアをフルスタックエンジニアと定義する。

3. フルスタックエンジニアに求めるスキル

前述の通り、フルスタックエンジニアにはクラウドの構成要素を網羅的に理解する必要がある。そのため、フルスタックエンジニアに求める IT スキルとして、以下の 7 つを定義する。

【フルスタックエンジニアに求める IT スキル】

① ネットワーク
② ストレージ
③ 仮想化
④ OS
⑤ DB
⑥ アプリケーション
⑦ セキュリティ

またアプリケーションをチームメンバーとともに構築することや、お客さまへ適切に説明する表現力がエンジニアには必要と考えるため、IT スキル 7 要素に社会人基礎力を加えた、全 8 要素をフルスタックエンジニアに求めるスキルと定義する。

【フルスタックエンジニアに求める追加スキル】

⑧ 社会人基礎力

【補足：社会人基礎力とは】

社会人基礎力：「1.前に踏み出す力（アクション）」、「2.考え抜く力（シンキング）」、「3.チームで働く力（チームワーク）」の 3 つの能力で構成する。

また、各能力は以下の能力要素から成る。

1．前に踏み出す力（アクション）
- a） 主体性
- b） 働きかけ力
- c） 実行力

2．考え抜く力（シンキング）
- a） 課題発見力
- b） 計画力
- c） 創造力

3．チームで働く力（チームワーク）
- a） 発信力
- b） 傾聴力
- c） 柔軟性
- d） 状況把握力
- e） 規律力
- f） ストレスコントロール力

経済産業省「社会人基礎力」http://www.meti.go.jp/policy/kisoryoku/

4. 新卒 3 年での姿

フルスタックエンジニアに将来なるために、新卒 3 年目で目指すべき姿を定義する。

【新卒 3 年目での姿】

クラウドで提供されているサービスを活用して、10 人月以下の Web アプリケーションが構築できる

①～⑦ については、研修や OJT、自己研鑽にてリテラシを養う。その後、クラウドを活用したアプリケーション開発を実践する。（アプリケーションについては設計手法、テスト手法を含む）

⑧については、研修や庶務業務、OJT を通じて育成する。

5. フルスタックエンジニアを育てるために

(1) 組織

① 育成専門組織の設立

入社初年度は育成のために開発に集中できる環境を提供することが重要だと考える。また、育成のためには周りの協力・理解も必要となる。そのためには育成をするための専門組織が必要になると考える。これまでのような新入社員の教育の一環として実施してきた電話対応や各種申請作業などの庶務業務や OJT の一環で実施している議事録作成などの業務は、新入社員にとって業務時間の大半を使うものであった。これらの作業は社会人基礎力を養う上では重要であると考えるため、なくす必要はないと考えるが、一日の業務時間のうち、数時間の開発時間中においては、上記の作業を忘れて開発に集中できる時間を提供したい。

一日の作業スケジュール（案）

9 時	10 時	11 時	12 時	13 時	14 時	15 時	16 時	17 時
予定の確認	開発①	開発②	昼食	開発③	庶務 OJT	庶務 OJT	報告書作成	振返り

② モチベーションの向上策を講じる

(ア)心理的安全性

心理的安全性とは、サイコロジカル・セーフティ(psychological safety)という英語を和訳したものであり、ビジネスと強い関連性を持つ心理学用語である。

心理的安全性は、他者の反応に怯えたり羞恥心を感じたりすることなく、自然体の自分を曝け出すことのできる環境や雰囲気のことを指す。プロジェクト活動などのビジネスシーンにおいても、本来の自分とは大きく異なる仕事用の人格を演じることなく、チームに所属する全てのメンバーが普段通りのリラックスした状態で活動に参加することを可能にしてくれる。

この心理的安全性という言葉自体は以前から存在していたが、アメリカの Google Inc.(以下、グーグル社)が 2012 年から約 4 年もの年月をかけて実施した大規模労働改革プロジェクト、プロジェクトアリストテレス(Project Aristotle)や、その他の人事関連研究の成果報告として『心理的安全性は成功するチームの構築に最も重要なものである』と発表した。

このように、エンジニアを育成していくためには心理的安全性を確保できる環境が必要である。

システム開発においても、仕様書（仕様書の行間含む）通り動かないため瑕疵にするといったことではなく、課題があっても徐々に改変するもの（もしくはすぐに改変できるもの）として前向きにシステム開発ができることが育成の段階では重要になると考える。

(イ)報酬

2017 年夏、中国の通信機器大手・華為技術（ファーウェイ）の求人情報がインターネットを騒がせた。同社の日本での初任給が 40 万円以上であると。

リクナビ 2018 に掲載されたファーウェイ・ジャパンの求人広告によると、募集職種は「通信ネットワークエンジニア」「端末テストエンジニア」「端末アフターサービスエンジニア」「研究職・アルゴリズムエンジニア」の 4 つ。月給は学士卒で 40 万 1000 円、修士卒で 43 万円に設定されている。年に 1 回以上は賞与があるというから、賞与が月給 2 か月分だとすると年収は初年度から 560 万円以上になる。

「有給消化 50%以上」「完全土日祝休み」ともあり、きちんと休むこともできるようだ。もちろん各種社会保険も完備されており、退職金制度も整っている

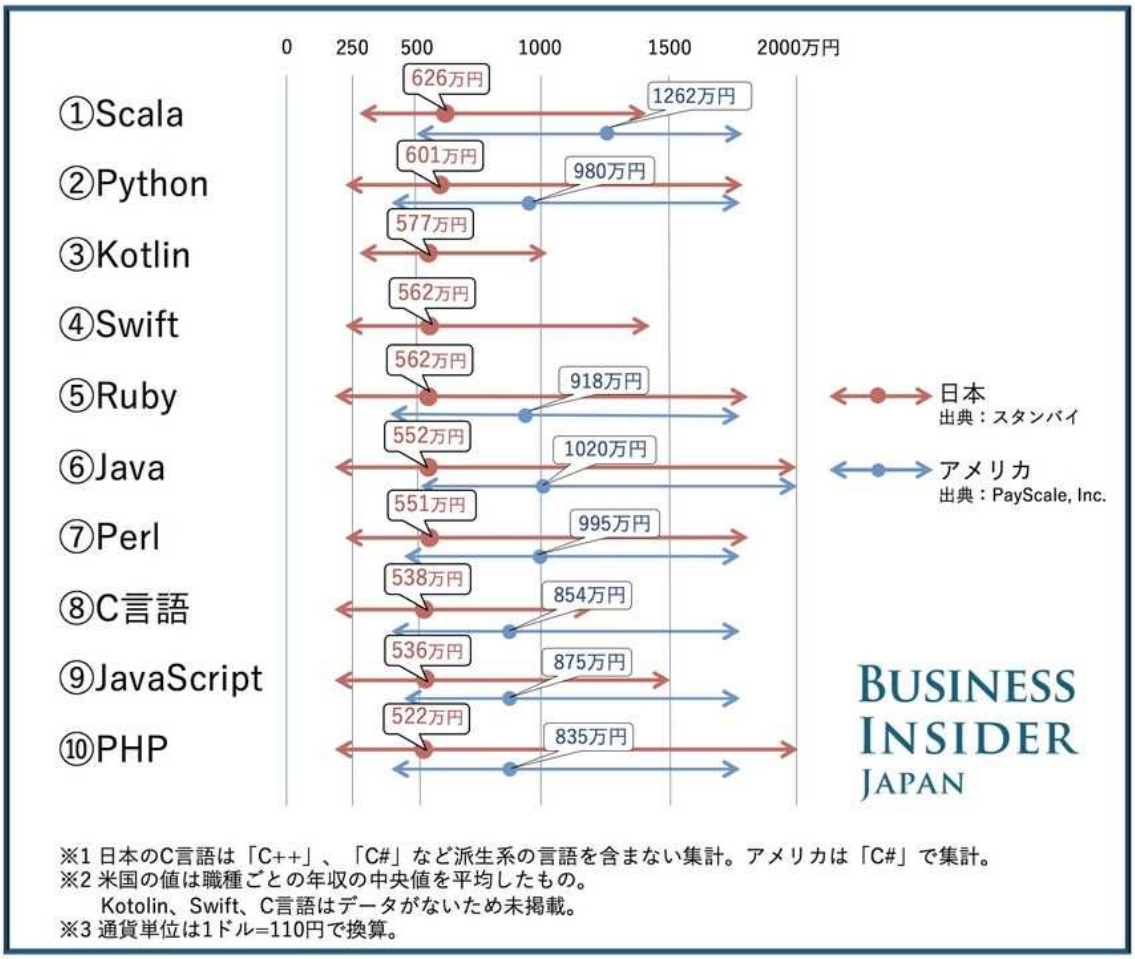
給与がすべてではないが、エンジニアとして自身の価値を計る指標として、給与が高いことにこしたことはない。

東京労働局の「平成 28 年 3 月新規学校卒業者の求人初任給調査結果」（http://tokyo-roudoukyoku.jsite.mhlw.go.jp/jirei_toukei/_121239/_121934.html）によると大学卒の初任給平均は 20 万 3000 円である。

前述のファーウェイは平均初任給の約 2 倍である。ただでさえ IT 人材不足という状況なのに、ファーウェイのような企業が出てくると優秀な人材が外資企業へ流れてしまう。

このような流れを食い止めるべく、IT 人材の価値を改めて見直す必要があると考える。

【参考】日米におけるプログラミング言語別年収調査



（出典）経済産業省「日米格差は 600 万円か？「プログラミング言語別 年収調査」の衝撃」（https://www.businessinsider.jp/post-100819）

(2) 人材育成

① 新卒 3 年目までの教育

新卒 3 年目で目指すべき姿になるための教育方針・計画を記載する。

i. 教育方針

IT に関する基本的な知識習得と実践する機会を提供する。

合わせて社会人としての基礎的な力を庶務業務や OJT を通じて身につける。

ii. 教育計画

(ア) IT スキル

[1 年目]

研修にて IT スキル定義の①～⑦について基本知識を習得する。そのうえで簡易な Web アプリケーション構築を実践し、プログラミングの実践やクラウドを活用したサーバ設定などを実践する。

[2 年目]

既存アプリケーションの改修を実践する。

[3 年目]

新規アプリケーションの設計からテストまでを実践する。

1～3 年での実施内容と IT スキルとのマップは以下の通り。

IT スキル			1 年目	2 年目	3 年目
1	ネットワーク				
1-1		リテラシ	○		
1-2		デバイス		○	
1-3		モバイル＆ソーシャル			○
2	ストレージ				
2-1		リテラシ	○		
2-2		RAID			○
2-3		バックアップ			○
3	仮想化				
3-1		リテラシ	○		○
3-2		サーバ&ストレージ			○
4	OS				
4-1		リテラシ	○		
4-2		Windows	○	○	○
4-3		UNIX/Linux	○	○	○

5	DB			
5-1	SQL	○		
5-2	DB 管理		○	○
5-3	DB 設計			○
6	アプリケーション			
6-1	基礎	○		
6-2	設計	○	○	○
6-3	実装	○	○	○
6-4	開発プロセス	○	○	○
7	セキュリティ			
7-1	リテラシ	○		
7-2	インフラストラクチャ			○
7-3	クラッキング			○
7-4	セキュリティポリシー		○	○

（イ）社会人基礎力

社会人基礎力については、庶務業務や OJT を通年で実施することで習得を目指す。

社会人基礎力		1 年目	2 年目	3 年目
1	前に踏み出す力（アクション）			
1-1	主体性	○		
1-2	働きかけ力			○
1-3	実行力	○	○	
2	考え抜く力（シンキング）			
2-1	課題発見力		○	○
2-2	計画力	○	○	
2-3	創造力			○
3	チームで働く力（チームワーク）			
3-1	発信力		○	○
3-2	傾聴力			○
3-3	柔軟性			○
3-4	状況把握力			○
3-5	規律力	○	○	
3-6	ストレスコントロール力		○	○

② 教育結果の評価方法

教育した結果、フルスタックエンジニアとして期待する姿になったのかを評価する方法を記載する。

フルスタックエンジニアに必要なITスキルと定義した項目を評価するアセスメントツールが存在する。GAITというスキルアセスメントツール（１７６問　９９０点満点）である。このツールを活用しITスキルの習熟度を評価する。（GAITを活用した事例を別紙4に記載する）

以下は、GAITの平均点とツールで評価できる分野を記載する。

【GAITの最高得点と平均点】

社会人の最高得点　：８９３点　　学生の最高得点　：７９９点
平均点　　：４２５点　　平均点　　：３３５点

【GAITの評価分野】

① ネットワーク
(ア)リテラシ
(イ)デバイス
(ウ)モバイル&ソーシャル

② ストレージ
(ア)リテラシ
(イ)RAID
(ウ)バックアップ

③ 仮想化
(ア)リテラシ
(イ)サーバ&ストレージ

④ OS
(ア)リテラシ
(イ)Windows
(ウ)UNIX/Linux

⑤ DB
(ア)SQL
(イ)DB管理
(ウ)DB設計

⑥ アプリケーション
(ア)基礎
(イ)設計
(ウ)実装
(エ)開発プロセス

⑦　セキュリティ
(ア) リテラシ
(イ) インフラストラクチャ
(ウ) クラッキング
(エ) セキュリティポリシー

このツールを入社時と研修後など随時活用し、個人の習熟度を評価する。また、都度評価することで、個人の自己研鑽習慣を得ることも副次効果として期待する。

③　3 年目以降の育成

3 年目以降はもはや育成ではなく、これまで得た知識を活かすとともに常に最新の知識を得つつ、実案件を実施し経験値を得ることを実施する。

フルスタックエンジニアとしてさらに成長するためには、実務経験を増やす必要がある。

(3) 自己研鑽

これまで育成方針について述べてきたが、被育成者自身がこの分野において技術に興味を持ち、自身で知識を得ることが大前提である。知識を習得するために努力を惜しまないこと。この前提がある者に対して、エンジニアとしての成長を促す環境と機会を私たちは提供しなければならない。

そうすることで、フルスタックエンジニアに求める素養を持ったエンジニアはおのずと成長するだろう。

以上

＜事例＞
フルスタックエンジニア
を目指すための
人財育成プログラム

N社取り組み事例より

プログラムの目的、目標

Foresight in sight

目的：
フルスタックエンジニアのベースとなるスキルの習得
異なる経験分野からのスキルチェンジ

育成目標ゴール	レベル		
	0	1	2
到達レベル定義	下記のレベル１，２に満たない	現場周辺、もしくは配下の担当として職場に配置できる資質とモチベーションを有し、システムエンジニアとしての基礎スキルを習得	現場に即配置できる資質とモチベーションを有し、システムエンジニアとしての即戦力となりうるスキルを習得

３つの能力形成指標（クライテリア）の観点

マインド・姿勢

テクニカルスキル（専門性別）

テクニカルスキル（共通）・ビジネス基礎・ヒューマンスキル

プログラムの構成

Foresight in sight

BasicとAdvanceの２フェーズで構成。
全員がBasicプログラムに参加後に、アイデンティティを確立できたか、Basic卒業クライテリアを満たしているかの結果を元に、Advanceフェーズへ進むか判断

レベル0→レベル1へ

Basic

マインド・姿勢・モチベーション

- 成長目標設定・アクションプラン策定
- 適性アセスメントⅠ
- 面談＜3ヶ月ごと＞と面談記録
- ワークショップ

ビジネス基礎・テクニカルスキル・ヒューマンスキル

- 基礎知識学習（eL・書籍・社内Web）
- 応用トレーニング（研修・課題）

4月～9月

レベル1→レベル２へ

Advance

マインド・姿勢

- メンタリング
- 適性アセスメントⅡ
- 面談＜3ヶ月ごと＞と面談記録
- 現場配属前研修

テクニカルスキル（専門性別）

- 基礎知識学習（eL・書籍・社内Web）
- 応用トレーニング（ワークショップ・課題）

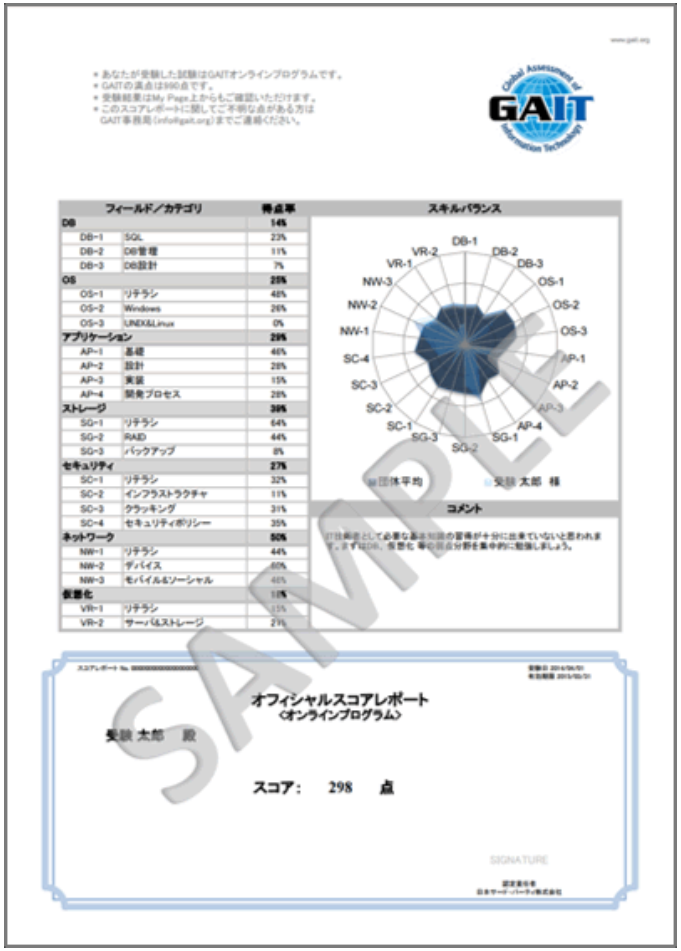
10月～3月

GAIT

Foresight in sight

IT全般の知識と経験を評価するアセスメントツール「GAIT」

- **クラウド時代に必要な技術要素7分野網羅 OS/NW/DB/AP/ストレージ/セキュリティ/仮想化**
- **出題数　176問、試験時間　65分、受験料　8,500円**



600点以上	IT技術者として十分な知識力がある
460点以上600点未満	IT技術者として基礎的な知識の理解ができている
350点以上460点未満	IT技術者として技術概念の理解ができている
250点以上350点未満	IT技術者としての基礎レベルに到達するまであと一歩
250点未満	IT技術者として必要な知識の習得がまだ不十分

GAITではブロンズ300～479　シルバー480～699　ゴールド700～990として認定

https://www.gait.org/business/casestudy/mki/

eL教材① ITエンジニアコアスキルでベースをつくる

Foresight in sight

必須

ITエンジニアコアスキル e-Learning (日本サードパーティ製)

- GAIT(クラウド時代に必要な技術要素7分野）を網羅した全65コンテンツ／17時間
- 価格　17,500円／１ユーザ　90日間視聴可能

技術要素	カテゴリ	コンテンツ	時間（目安）
OS	①リテラシ ②Windows ③UNIX／Linux	①OSの概要／OSの役割／OSの種類 ②インストール／設定／管理 ③インストール／設定／管理	2.5時間
NW	①リテラシ ②デバイス ③モバイル／ソーシャル	① ネットワークの概要／機能とプロトコル／インターネットアプリケーション ②デバイス1／ デバイス2／ 設計と構築 ③モバイル／ソーシャルネットワーク	2時間
DB	①SQL ②DB管理 ③DB設計	①DML／DDL／DCL ② 概要と導入／チューニング／運用 ③概念設計／論理設計/物理設計	2.5時間
AP	①基礎 ②設計 ③実装 ④開発プロセス	①ソフトウェアの利用／ソフトウェアの設定／ソフトウェアの導入 ②ソフトウェア品質／ソフトウェア開発の役割／表記法 ③データフォーマット／プログラミング／ソフトウェア開発環境 ④開発プロセスの種類／テスト技法／アルゴリズム	3時間
ストレージ	①リテラシ ②RAID ③バックアップ	①ハードウェア／設計と構築／運用と保守 ②RAID概要／設計と構築／運用と保守 ③バックアップ概要／設計と構築／運用と保守	2.5時間
セキュリティ	①リテラシ ②インフラストラクチャ ③クラッキング ④セキュリティポリシー	①セキュリティ防御手法／暗号化技術／情報セキュリティの概念 ②情報セキュリティインフラストラクチャ／認証技術／ネットワーク機器のセキュリティ／暗号化技術の応用／セキュアプロトコル ③セキュリティ攻撃手法 ④組織レベルのセキュリティ／個人レベルのセキュリティ／ネチケット	3時間
仮想化	①リテラシ ②サーバ＆ストレージ	①概要／アーキテクチャ／クラウドコンピューティング ②設計／構築／運用・管理	1.5時間

eL教材② CampのE-Learning　ITスキルで補足・深堀する

Foresight in sight

- 前頁のITエンジニアコアスキルを補足・深堀するために、以下 eLから各自選択

選択

技術要素	社内Camp　e-Learningコンテンツ	時間（目安）
OS	Windows Server の導入と管理 Windowsとの比較で学ぶLinux基礎	3時間 5時間
NW	TCP/IPネットワーキング基礎 ネットワーク設計基礎	2.5時間 3.5時間
DB	SQL入門 SE必須知識　データベース技術基礎	8時間 5時間
AP	Webアプリケーション基礎技術 Webアプリケーションセキュリティ基礎 SE必須知識　ソフトウェアテストの基礎技術 ソフトウェア品質基礎 ソフトウェア構成管理の基礎技術 UML基礎 Javaプログラミング基礎 C#プログラミング Visual Basicプログラミング基礎（.NET対応） Cプログラミング基礎 COBOLプログラミング基礎 Office2003ユーザのためのoffice2013	2時間 2時間 12時間 1時間 2.5時間 3時間 24時間 24時間 24時間 21時間 18時間 3時間
ストレージ	ストレージ基礎	3時間
セキュリティ	情報セキュリティ概論 ネットワークセキュリティ基礎 情報セキュリティ基礎	1時間 4時間 3.5時間
仮想化	該当なし	

上記は一部であり、他に多数コースあり

eL教材③ CampのE-Learning PM／情報リテラシのベースをつくる

Foresight in sight

- PM関連・情報リテラシ強化として、以下 eLから各自選択

選択

技術要素	社内Camp e-Learningコンテンツ	時間（目安）
プロジェクトマネジメント	プロジェクト･スコープ･マネジメントのツールと技法	3時間
	プロジェクト･タイム･マネジメントのツールと技法	5時間
	プロジェクト･コスト･マネジメントのツールと技法	3時間
	プロジェクト品質マネジメントのツールと技法	3時間
	プロジェクト人的資源マネジメントのツールと技法	3時間
	プロジェクト･コミュニケーション･マネジメントのツールと技法	3時間
	プロジェクト･リスク･マネジメントのツールと技法	5時間
	プロジェクト調達マネジメントのツールと技法	3時間
情報リテラシ	めざせ！ 業務の達人 - Excelマクロで作業を効率化（基礎編） -	3時間
	めざせ！ 業務の達人 - Excelマクロで作業をさらに効率化（活用編） -	3時間
	Office2003ユーザのためのOffice2010	3時間
	Office2003ユーザのためのOffice2013	3時間
	Excel2013入門	6時間
	Excel2013応用	6時間
	Word2013入門	6時間
	PowerPoint2013入門	6時間

GAIT結果

Foresight in sight

受験者22名(76%)の要員が得点アップ。29名のスコア平均は430点・30点アップ

得点順スコア

8月スコア	11月スコア	アップ点	アップ率
625	714	89	114%
519	539	20	104%
451	536	85	119%
363	526	163	145%
503	521	18	104%
440	519	79	118%
466	513	47	110%
367	485	118	132%
434	480	46	111%
398	461	63	116%
449	459	10	102%
425	456	31	107%
428	451	23	105%
408	438	30	107%
469	430	-39	92%
484	426	-58	88%
344	404	60	117%
335	404	69	121%
422	394	-28	93%
346	394	48	114%
418	382	-36	91%
234	371	137	159%
344	357	13	104%
360	344	-16	96%
301	343	42	114%
419	338	-81	81%
295	300	5	102%
258	269	11	104%
324	215	-109	66%
401	430	29	108%

参考：業界別平均スコア　2016年1月～6月

社会人　最高点　776

ITコンサルティング　656

パッケージソフトウエア開発　494

ユーザー系システムサービス　432

システムインテグレータ（Sier）　430

平均点　401／社会人

出典：GAIThttps://www.gait.org/about-gait/result-fy16-1st/

プログラムの評価

Foresight in sight

評価：

- **１年間の当該プログラム受講でフルスタックエンジニアのベースとなるスキル習得は可能（GAIT得点の向上）**
- **ティーチング、メンタリングを組み合わせることで学習者が自立的に学習**
- **異なる経験分野からのスキルチェンジについて自信をもたせることができた**
- **テクニカルスキルだけの習得ではなく、マインド形成、ビジネススキル、ヒューマンスキルについても平行して訓練することで現場への適用力がUP**

３つの能力形成指標（クライテリア）の観点