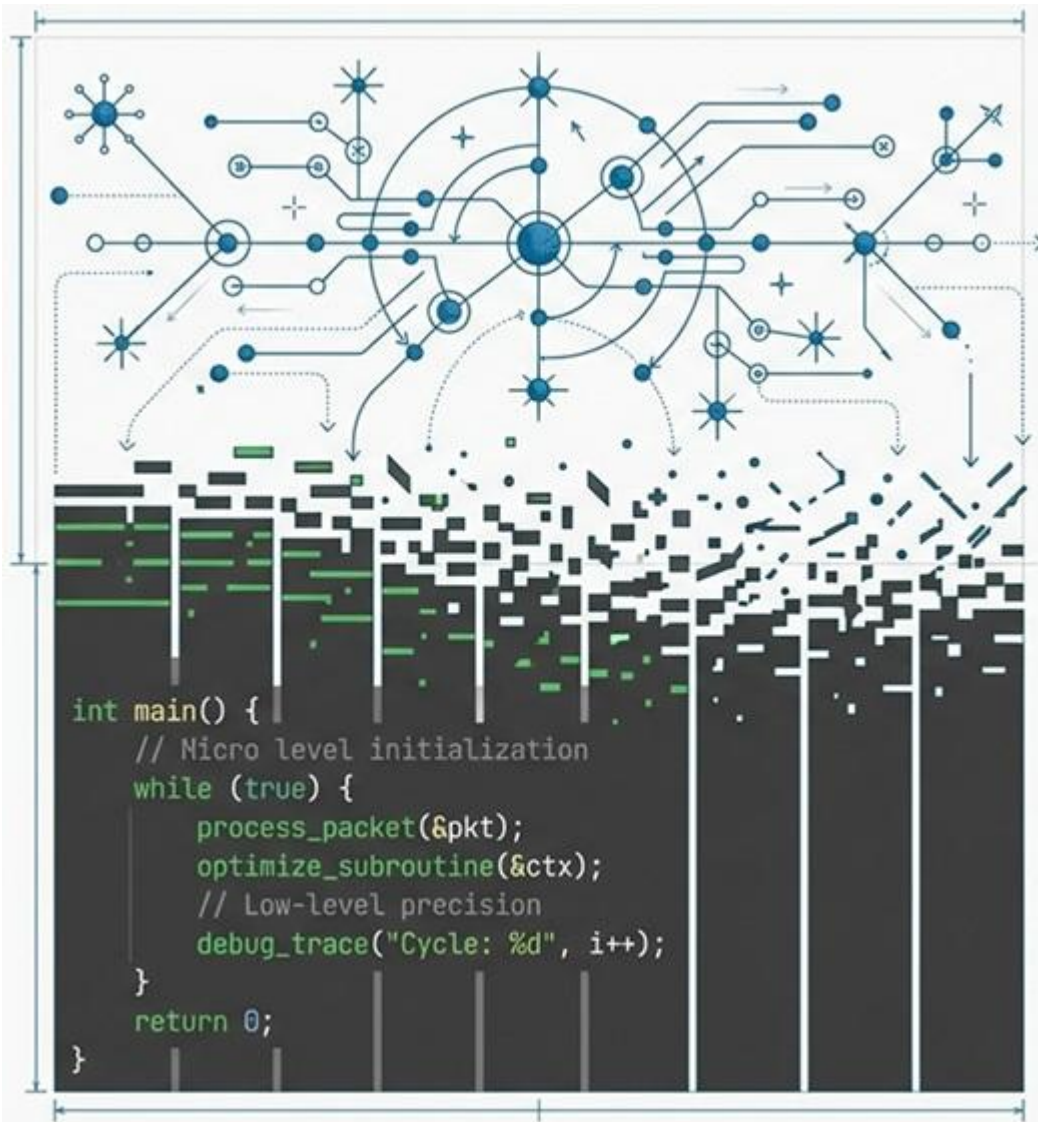


技術者にとっての プライドとはなにか

孤高の職人から、知のオーケストラへの進化

CITP 18001042
新村 幸裕



```
int main() {  
    // Micro level initialization  
    while (true) {  
        process_packet(&pkt);  
        optimize_subroutine(&ctx);  
        // Low-level precision  
        debug_trace("Cycle: %d", i++);  
    }  
    return 0;  
}
```

はじめに

- 資料および発言の内容は個人の見解に基づくものであり、所属組織を代表するものではありません。
- 本資料は、NotebookLMにより生成された内容をもとに加筆修正したものとなります。



孤高な技術者、 私は好きです

自身の技術に誇りを持ち、自らの道を歩む技術者。
かっこいいと思います。

しかし、課題が複雑化し、技術進歩が加速する
現代において、たった一人ですべてを解決すること
は限界を迎えていると考えています。

自己紹介

情報システムエンジニア、プロマネ、DXコンサル

- ・ 通信系システムなどミッションクリティカル系
ソフトウェア開発（Linux、C言語、システム冗長方式）
プロジェクトマネジメント
- ・ 新規事業開拓（通信系やセキュリティ系）
- ・ DXによる社会イノベーション（DXコンサル）

生々しい現場の悦び

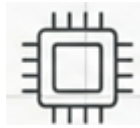
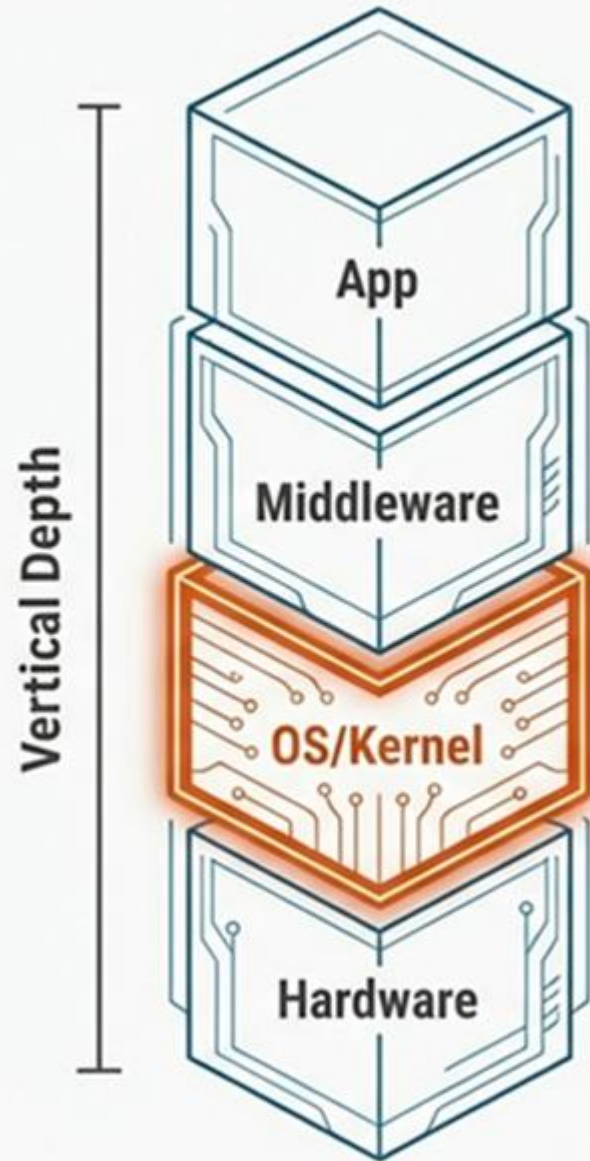
Linux tcp.c 洗練されたソフトウェア

```
int tcp_setsockopt(struct sock *sk, int level, int optname, sockptr_t optval,
                  unsigned int optlen)
{
    const struct inet_connection_sock *icsk = inet_csk(sk);

    if (level != SOL_TCP)
        /* Paired with WRITE_ONCE() in do_ipv6_setsockopt() and tcp_v6_connect() */
        return READ_ONCE(icsk->icsk_af_ops)->setsockopt(sk, level, optname,
                                                         optval, optlen);
    return do_tcp_setsockopt(sk, level, optname, optval, optlen);
}
```

生々しい現場の悦び

ソフトウェア開発の核心 （私の場合）



C言語の世界

メモリワーク、ポインタ、構造体



排他制御の妙

いかに待ちをなくし、リソース競合を避けるか。



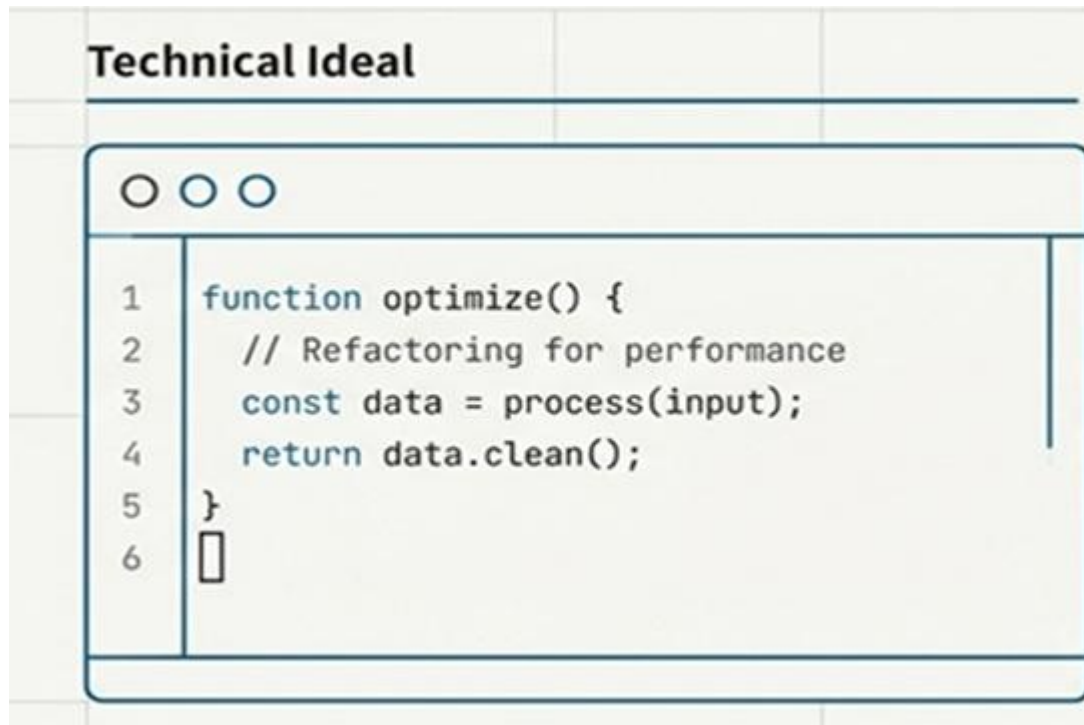
トラブルシューティング

メモリリーク、デッドロック、printf(), アセンブラ

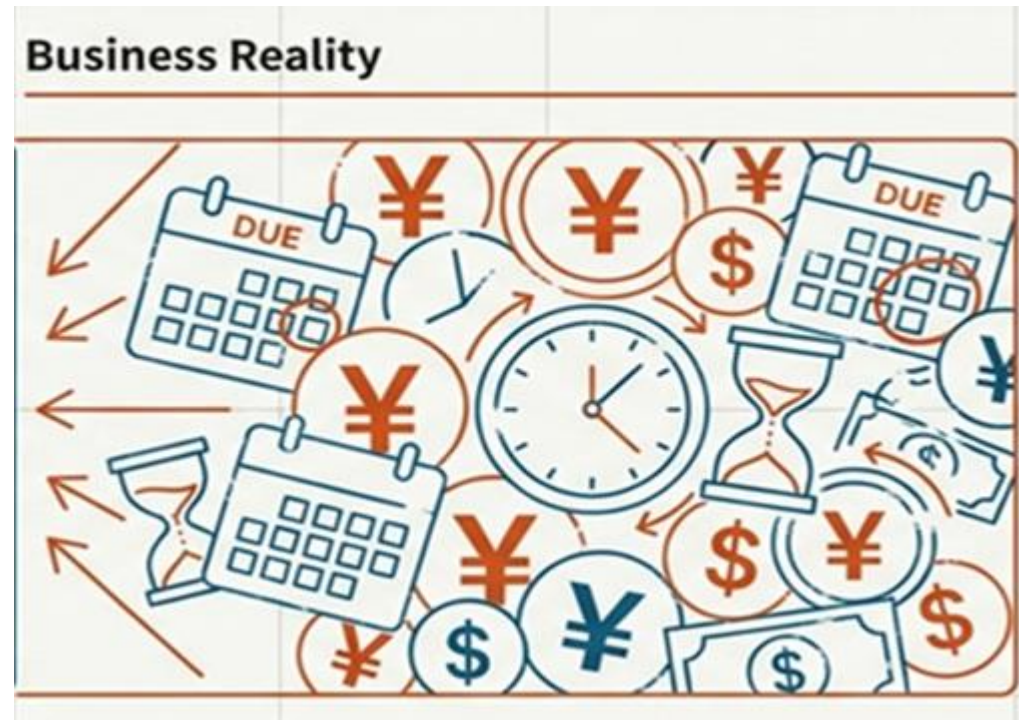
Takeaway

いかに安全で高機能なものを、効率よく作るか。
この「深堀り」こそが技術者としての楽しみ。

「無限の工数」は存在しない



ソフトウェア開発手法やメトリクスを意識し始めると、
他流派（他社・他業種）が気になり始めます。

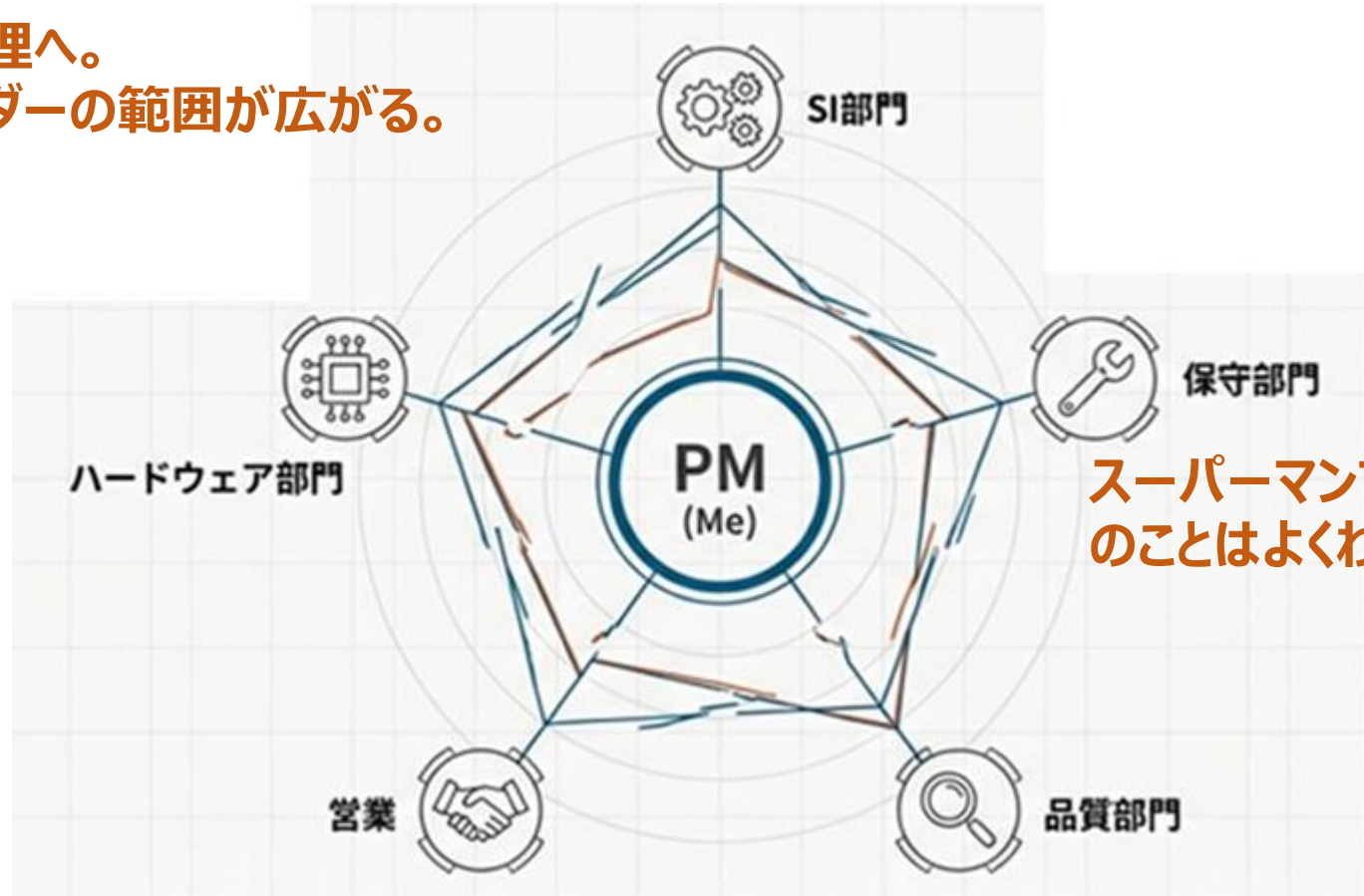


企業である以上、納期もあれば
コストの限界もあります。

技術的な理想とビジネス的な制約。他が気になり、視野は強制的に広げられた。

プロマネという名の「調整」の海へ

開発から、企画・管理へ。
一気にステークホルダーの範囲が広がる。



スーパーマンではないので、他の専門領域のことはよくわからない・・・

みなそれぞれに思いと主張がある。「よい塩梅」を見つける難しさ・・・

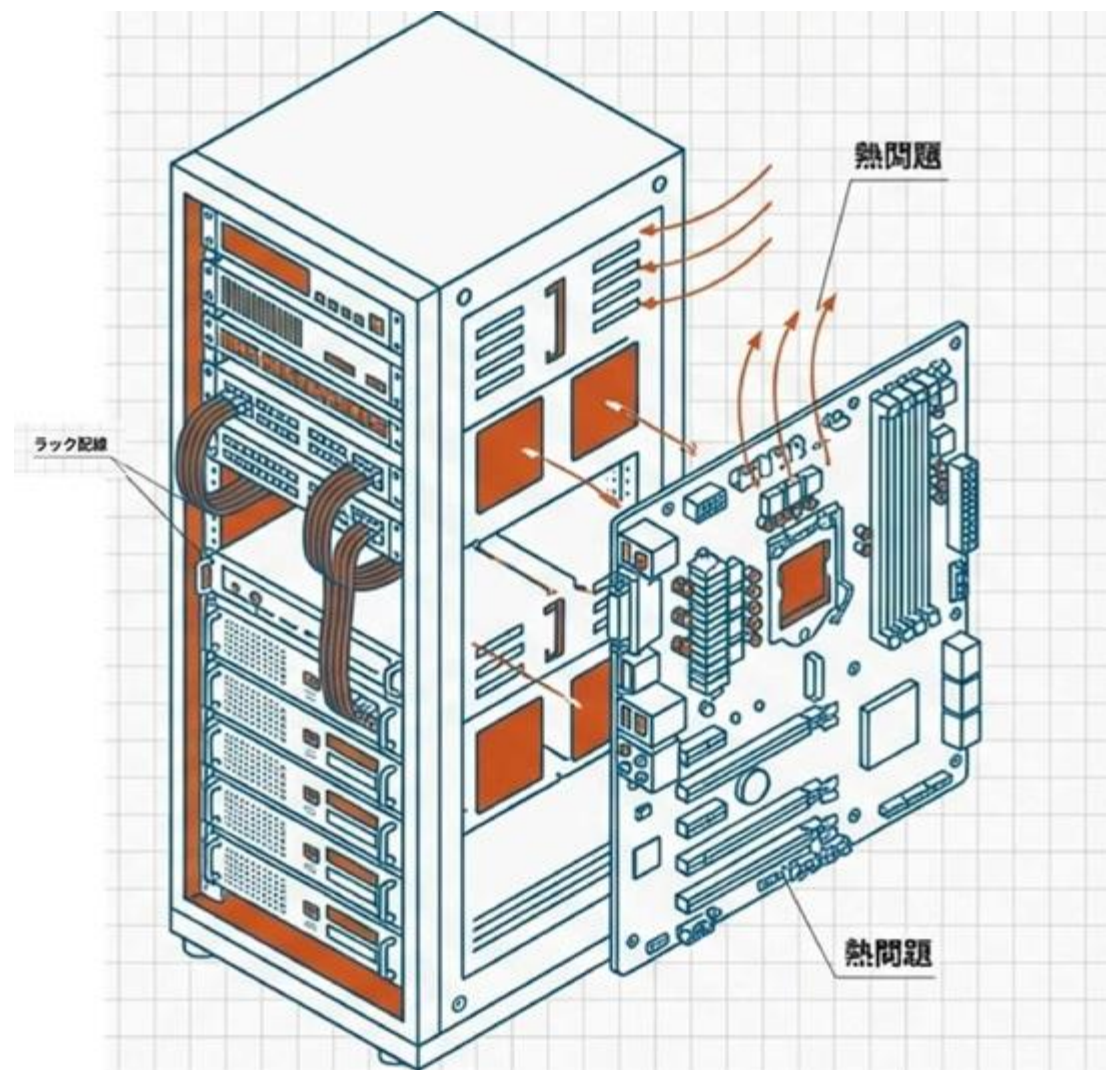
未知を知る 楽しさと、限界

隙間時間でのハードウェア修行

回路設計、ラック配線、熱問題、太陽フレアの影響...

知識が広がる楽しさを知る一方で、気づくこと。

「一人で全部背負い込まなくていい。
有識者と連携すればいいのだ」と。



知識が広がる楽しさを知る一方で、気づくこと。

「一人で全部背負い込まなくていい。
有識者と連携すればいいのだ」と。

プロマネとして、いろいろな人を巻き込むが、、

私は、まだ、技術者なのか？

ただ人を巻き込むだけ？

人の知識を仕入れてくるだけ？

ただの調整役になってしまったのではないか？

ある時、

私は、まだ、技術者なのか？

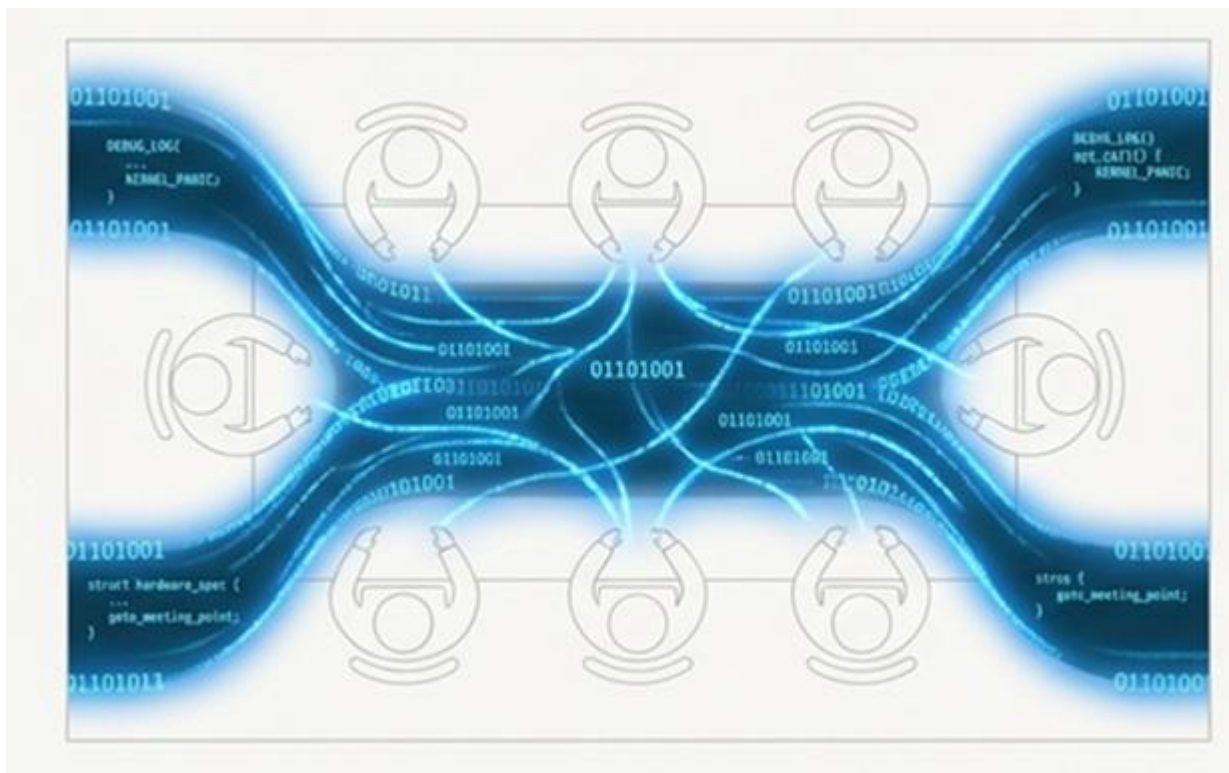
ただ人を巻き込むだけ？

人の知識を仕入れてくるだけ？

ただの調整役になってしまったのではないか？

**原因不明のバグへ応急処置（ふたをする）、強烈な「気持ち悪さ」を感じた。
技術者としてのプライドの残り火でした。私は技術者。**

泥臭い経験が、武器になる



複雑な問題が発生した時、
リーダーでも率先して、

ソースコードを解析、
カーネル(OS)を追い、
テストコードを埋め込む。

Key Insight

言語仕様、ソフト構造、ハードウェア特性、
これらを知っている技術者だからこそ、
問題を「俯瞰的」に捉え、
適切なステークホルダーを巻き込める。

プロを動かすには

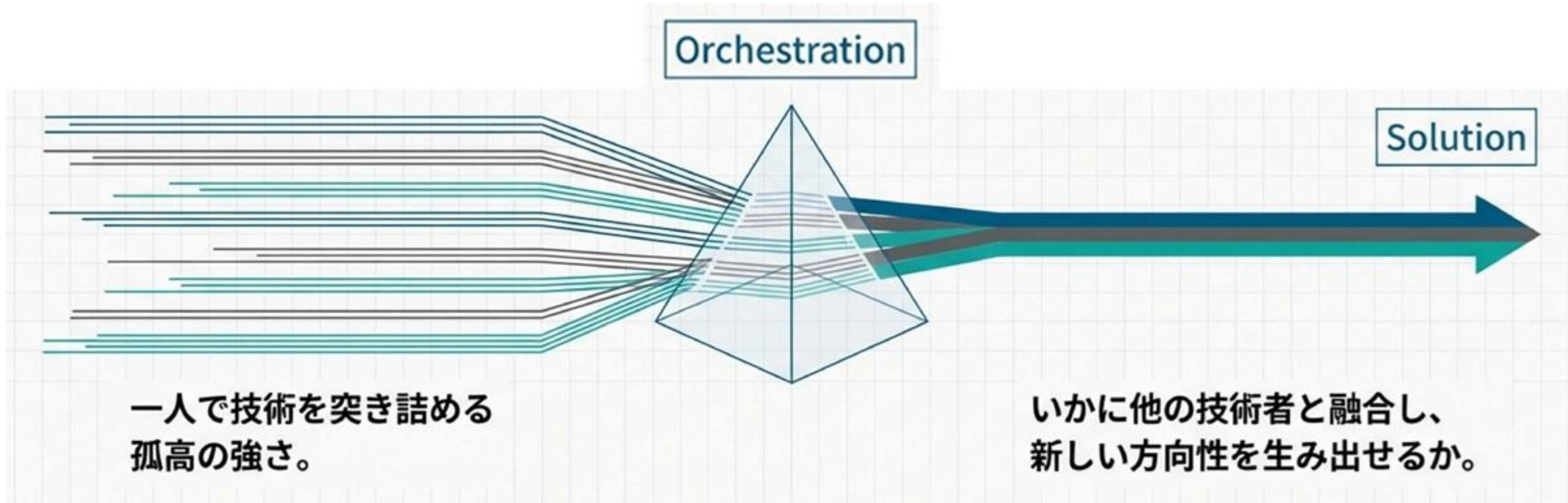
**「他分野のプロを巻き込むには、
自身も、ある分野のプロでないといけない」**

自分の専門領域に自信があるからこそ、他者の専門領域に対する「敬意」が生まれる。



**中途半端な知識では、
他のプロを認め、受け入れることはできない。**

新しい「技術者のプライド」



「みなで技術を結集し、難課題の解決、新しい発見」

達成感、喜び、楽しさ

これができる！

複雑な世の中を生き抜くための3つの行動



まずは聞く

他分野であっても、
敬遠せずに耳を傾ける。



柔軟な思考と応用

自身が解決しようとして
いる課題に、他分野の知
恵を当てはめる。



融合を楽しむ

思いもしなかったより良
い方向性が見出せること
に、喜びを見出す。

新しいことを受け入れる柔軟性

自己革新組織

p196

「組織の**自己革新は学習だけではできない**。学習には、絶えず過剰適応の危険が伴うからである。環境適応に有用であった知識のみを選択し蓄積していくと、**「過去の成功体験への過剰適応」**が起こり、新たな環境変化に適応できなくなるのである（戸部良一他『失敗の本質』）。すなわち自己革新組織は、主体的に**新たな知識を創造**しながら、既存の知識を部分的に棄却あるいは再構築して自らの知識体系を革新していくのである。この意味で知識創造こそが組織の自己革新の本質なのであり、**新しい知の創造なくして組織の自己革新はあり得ない**のである。」

p199

「組織の自己革新にとって結晶化のもつ本質的な危険性である。結晶化とは、概念が具体的なモノあるいはシステムに結実することであるが、その実現過程でかかった**費用や努力、そしてそれに伴う思い出や愛着心**を考えると、たとえそれが環境の変化に伴い本来意図した機能を果たさなくなったとしても、**いったんできあがったものを壊すことは個人的にも組織的にもきわめて困難**である。」

		
まずは聞く 他分野であっても、 敬遠せずに耳を傾ける。	柔軟な思考と応用 自身が解決しようとして いる課題に、他分野の知 恵を当てはめる。	融合を楽しむ 思いもなかったより良 い方向性が見出せるこ とに、喜びを見出す。



野中郁次郎, アメリカ海兵隊, 中公新書, 2025

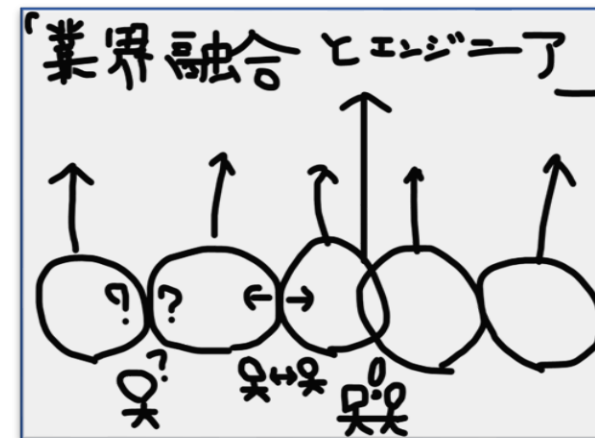
人、本音、広く多様な視野

		
まずは聞く 他分野であっても、 敬遠せずに耳を傾ける。	柔軟な思考と応用 自身が解決しようとして いる課題に、他分野の知 恵を当てはめる。	融合を楽しむ 思いもなかったより良 い方向性が見出せること に、喜びを見出す。

✓ **SSM(ソフトシステム方法論**：ピーター・チェックランド氏) 2016年研修
技術者（エンジニア）としての考え方が変わったきっかけ

やはり、人、ゆえに本音

融合には、他メンバとの本音のすり合わせ、腹おちが大事



✓ **幅広い領域での思考**

労働価値（資本論：カール・マルクス氏）

最近のAI革命

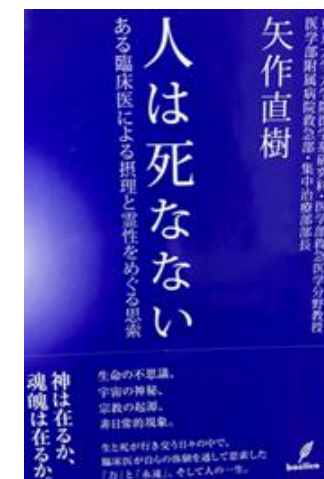
そういえば、と、資本論。

現代と合わないところはあるかも知れないが、まさに今起こっているAI革新と重なる点も感じた。

（AIにより労働価値が下がってしまうと仮定 → 雇用縮小、賃金停滞 ？）

（技術者は、経済的な視点からも技術のあるべき方向を見なければいけない。高ハードル。）

他技術者に限らず、他分野の専門家との融合も重要。



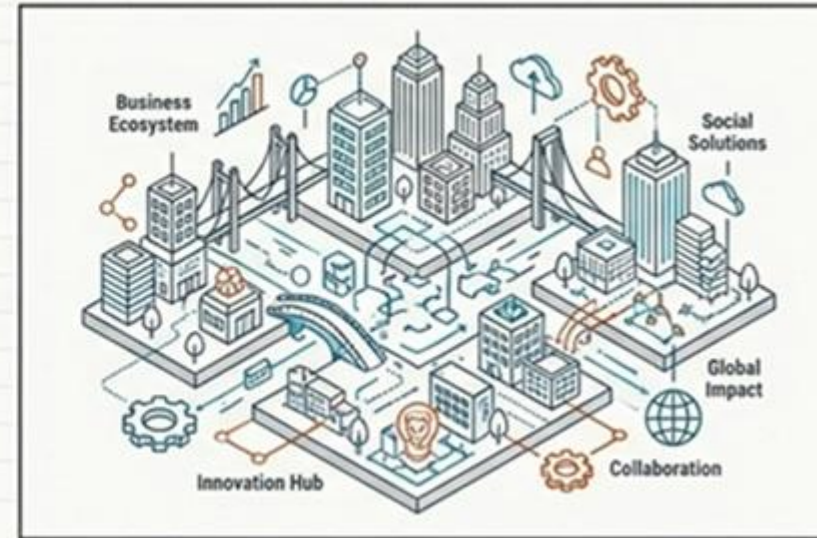
✓ CITPなどのコミュニティでは、このような**未経験ゾーン**にも出会える（摂理と霊性）

矢作直樹. 人は死なない. バジリコ, 2013

根本は変わらず

ソフトウェア開発から、ビジネス創生へ

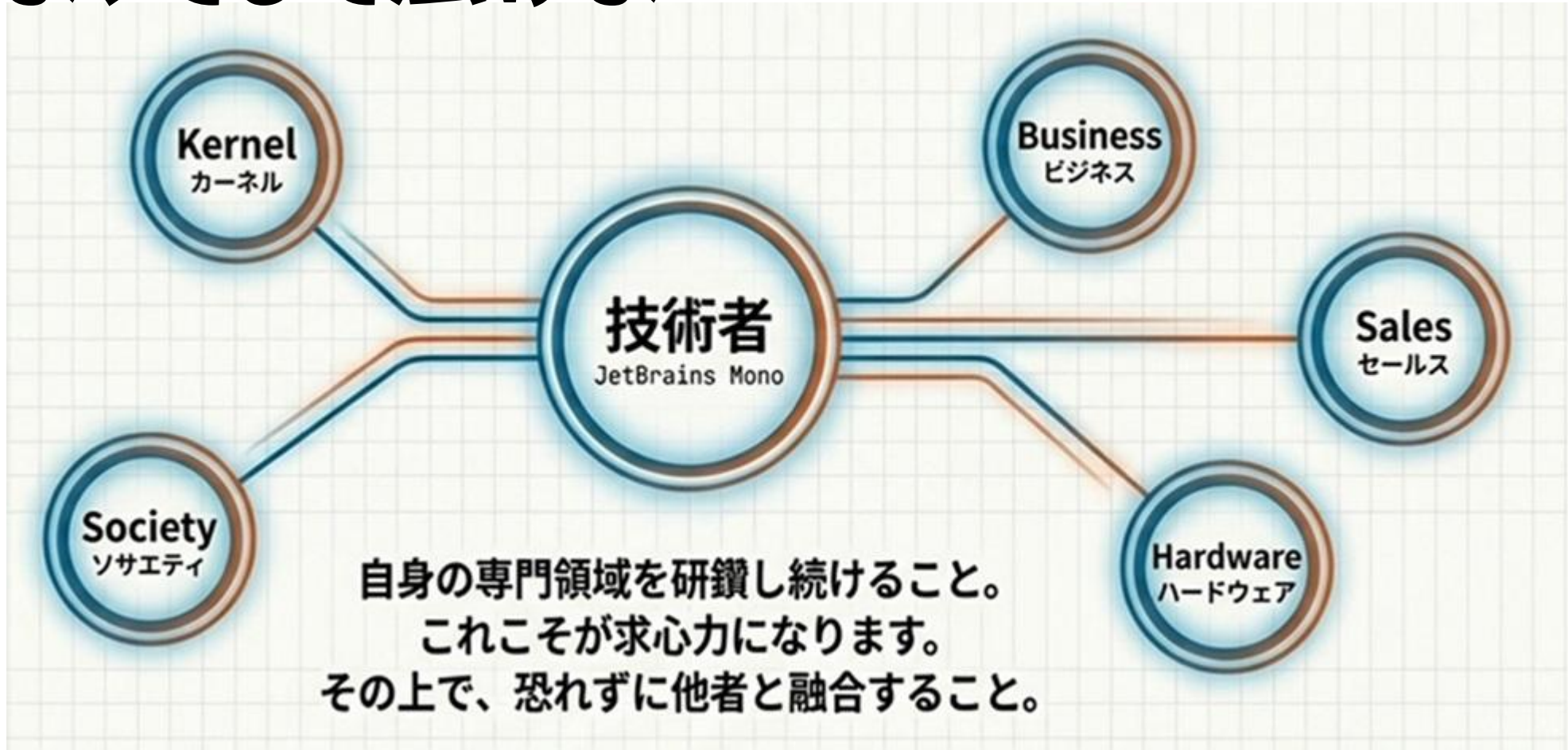
```
int main() {  
    // Initialize complex variables  
    VariableSet complex_vars = load_data();  
    // Transform and analyze  
    Result output = solve(complex_vars);  
    // Output solution  
    return output.status;  
}
```



ソフトウェア開発の中心からは離れ、ビジネス創生や社会課題の解決に取り組んでいます。
しかし、根本は変わりません。

「技術者としての心構え」をフル活用し、
様々な領域のプロフェッショナルと連携して、
複雑な変数を解き明かす。

深めよ、そして広げよ



自己研鑽、柔軟な他者との融合
これこそ、現代における「技術者のプライド」ではないでしょうか？

あなたは、どう思いますか？

Thank you for your attention.