

特定掲示板「ソフトウェアの法則」より抜粋

法則1 【哀しきプログラマ】 虫を作るプログラマは、デバッグ作業を熱心によく働き者であると見なされる。 虫を作らないプログラマは、デバッグ作業を熱心にやらない怠け者であると見なされる。 【プログラマの決断】 プログラマは、 1. 決断を人にまかせると、依頼心が強いと思われる。 2. 決断を成り行きに任せると、優柔不断と言われる。 3. たまに決断を下すと、最悪の選択をする。
法則2 【プログラマに対する評価】 プログラマの給与は、能力に反比例する。
法則3 【プログラマのジレンマ】 平凡なコードには似てしまう。非凡なコードには似せなければならない。 非凡なプログラムには、似ないように似せなければならない。
法則5 【アイデアの不確定性原理】 1. 何かアイデアを思いつくと、必ずメモ用紙が手近にない。 2. メモ用紙の代わりにやっとみつけると、鉛筆がない。 3. 鉛筆が見つかったときには、アイデアは思い出せない。 【アイデアに対するニコチン効果】 ニコチン中毒のプログラマは、 1. 時として、良いアイデアが浮かんだと確信する。 2. 一服やれば、そのアイデアがまとまるような気がする。 3. 喫煙エリアに急ぐ間にアイデアは、はかなくも忘れ去られる。 【ニコチン効果に対するコメント】 プログラマは一服やると、そのアイデアが錯覚であったことに気がつく。 【ニコチン効果に対するコメントへの修正】 プログラマは、喫煙エリアに急ぐ間にアイデアが消えたと思いたがる (もともとアイデアなど思い浮かばなかったのに)。
法則6 【軽度の心身症の診断法】 世の中のすべてが自分を中心にして回っているように感じられたら、早く家に帰って休養した方がよい。 【強度の心身症の診断法】 すべての人の視線が自分に殺到し自分の一挙手一投足が監視されているように感じられたら、これは極めて危険な状態である。従業員バッジをつけたまま電車に乗っている可能性が高い。
法則7 【マニュアルの分かりやすさ】 マニュアルの文章は、すでにその使い方を知っている人にしか理解できないような言い回しで記述されている。 【マニュアル批判の公理】 マニュアルに対するどんな悪口も、必ず正当化される。
法則9 【プログラマの夢】 楽観的プログラマの夢: "自動虫下しソフトウェア"の開発。 【プログラマの夢に対する修正】 楽観的プログラマの悩み: 自動虫下しソフトウェアのデバッグ法。
法則10 【プロジェクトリーダーの法則】 プロジェクトが苦しい局面に陥っても泰然としていられるリーダーは、すでに新たな転職先を見付けている。
法則11 【災難の法則】 ウィルスは忘れた頃にやって来る。虫は忘れないうちにやって来る。
法則14 【ウィンドウズ利用上の問題点】 ウィンドウズシステムは、机上作業のすべてを代替してくれる。しかし、マウスを転がす場所だけは提供してくれない。

法則15  
【結合則の怪】  
虫のあるものと虫のあるものを結合すると、虫のあるものになる。  
虫のないものと虫のあるものを結合すると、虫のあるものになる。  
虫のないものと虫のないものを結合すると、やはり虫のあるものになる。

法則16  
【デバッグの法則】  
どんなコードも最初は正しく見える。  
【デバッグの法則への反論】  
最初から殺人者のように見える犯人はいないが、最初から虫のように見えるコードは存在する。  
【不運なプログラマ】  
プログラムの虫には、コードの中に存在する虫と存在しない虫(コードの欠落)とがある。プログラマにとっての最大の不運は、存在しない虫に対してもコードの中を捜し回らなければならぬ点にある。

法則17  
【初体験の法則】  
プログラマは、  
1. 初めて作ったプログラムが正常に動作したときのことは決して忘れない。  
2. 最初に虫を作り込んだプログラムことは決して思い出せない。  
3. 両者が同一のプログラムであるという事実には決して思い至らない。

法則19  
【取説の書き方の法則】  
取説における「巧みな表現」とは、将来言い逃れができる余地を残した表現方法をとることである。  
【理解と誤解の法則】  
ソフトウェア説明書の文章は、決して作成者の意図通りには理解されない。必ず、恐れている通りに誤解され  
【誤解の法則】  
誤解を恐れずに書くと、その通りに誤解される。

法則24  
【トラブル対策上の基本的問題点】  
トラブル対策の責任を持つ人は、虫を修正する技術を持たない。虫を修正する技術を持っている人は、トラブル対策の責任を持たない。  
【謝罪の法則】  
誤りに行く人は、状況の深刻さを正確には把握していない人の方がよい。

法則26  
【設計変更に関する評価の法則】  
設計変更というものは、いつも最初は「たいした影響はない」ように見える。  
【要求仕様のはかなさ】  
正式版は、テスト版の通りには動かない。  
テスト版は、要求仕様の通りには動かない。  
故に、要求仕様と正式版とは何の関わりもない。

法則29  
【プログラムの単純化の法則】  
プログラムを複雑にするのは簡単で手間はかからないが、プログラムを簡単にするには複雑な手間がかかる。  
【シュリンクの法則】  
洗練されたプログラムと洗ったシャツは縮む。  
【拡大縮小の法則】  
シャツは洗うたびに小さくなるが、プログラムは改造するたびに大きくなる。

法則36  
【リバースエンジニアリングの法則】  
ハードウェアとは、組み立てるよりも分解する方がはるかに簡単であるものをいう。  
ソフトウェアとは、組み立てるのも分解するのと同じように簡単でないものをいう。  
【UNIXとは】  
UNIXとは、規格統一(Unification)を繰り返すが永遠に達成できない(X)システムの略称である。  
【ウィンドウズ利用上の注意】  
ウィンドウは換気が悪いので、ときどき開放(解放)して空気を入れ換えましょう。

法則39  
【スケジュール進捗に関する経験則】  
ソフトウェアの開発スケジュールは、納期の直前までは順調に進む。

法則40  
【ファイル保存の効用】  
人間は、二度と使う可能性のないファイルでさえもとりあえずハードディスク上に保存することによって心の安定を得る。  
【ファイルの保存則】  
ファイルは、永く保存しておけばおくほど熟成されて捨てがたくなる。それを捨てる決断がつくのはハードディスクが満杯になったときだけである。  
【ファイル破壊の予知法則】  
ファイル破壊の事故は、常にファイルを保存したいと思う直前に発生する。

法則42

【運不運の法則】

簡単に直るトラブルの発生源は自分の担当モジュールではない。

たちの悪い致命的なトラブルの発生源は、どういうわけか必ず自分の担当モジュールである。

【自然治癒の法則】

プログラムの熱い期待にもかかわらず、虫が自然に直ることはない。

【虫の発見に関する法則】

捜していない虫はすぐ見つかる。

しかも、致命的なトラブルでとりこんでいる最中に。

法則46

【バージョンアップの法則】

バージョンアップ・サービスとは、有料で欠陥商品を回収する手法のことである。

法則49

【ドキュメントの法則】

プログラマは必ず、前任者がドキュメントを残さなかったと嘆く。

法則50

【たちの悪い虫の法則】

たちの悪い虫は、必ず重要顧客で発生する。

法則53

【補充の法則】

遅れているプロジェクトに人員を補充すると、スケジュールはさらに遅れる。

補充しないで放置すると、プロジェクトは壊滅する。

【見積りの精度に関する経験則】

川の深さは、実際に渡ってみなければ分からない。川を渡るのに要する時間も、渡ってみなければ分からない。