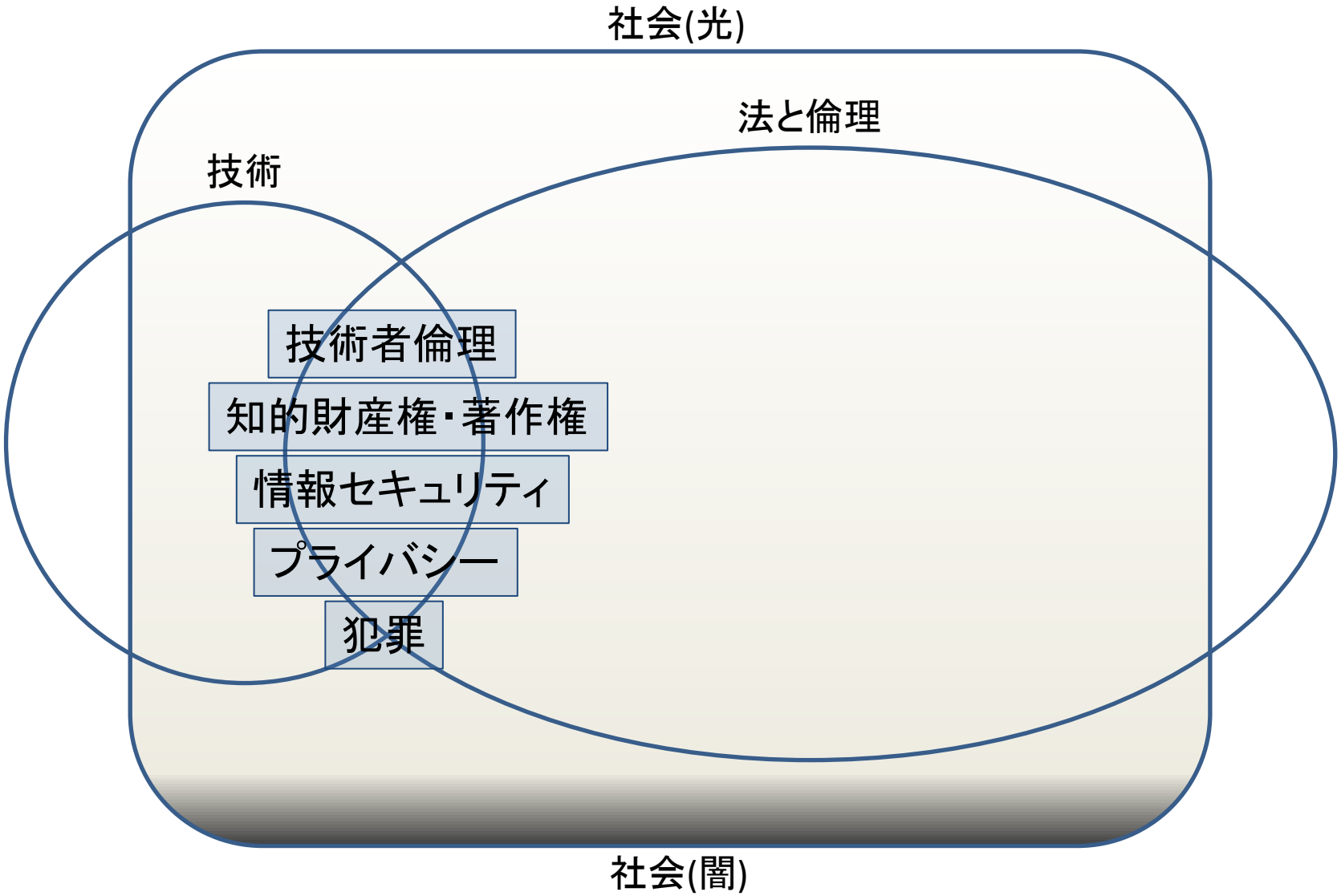
情報セキュリティと情報倫理

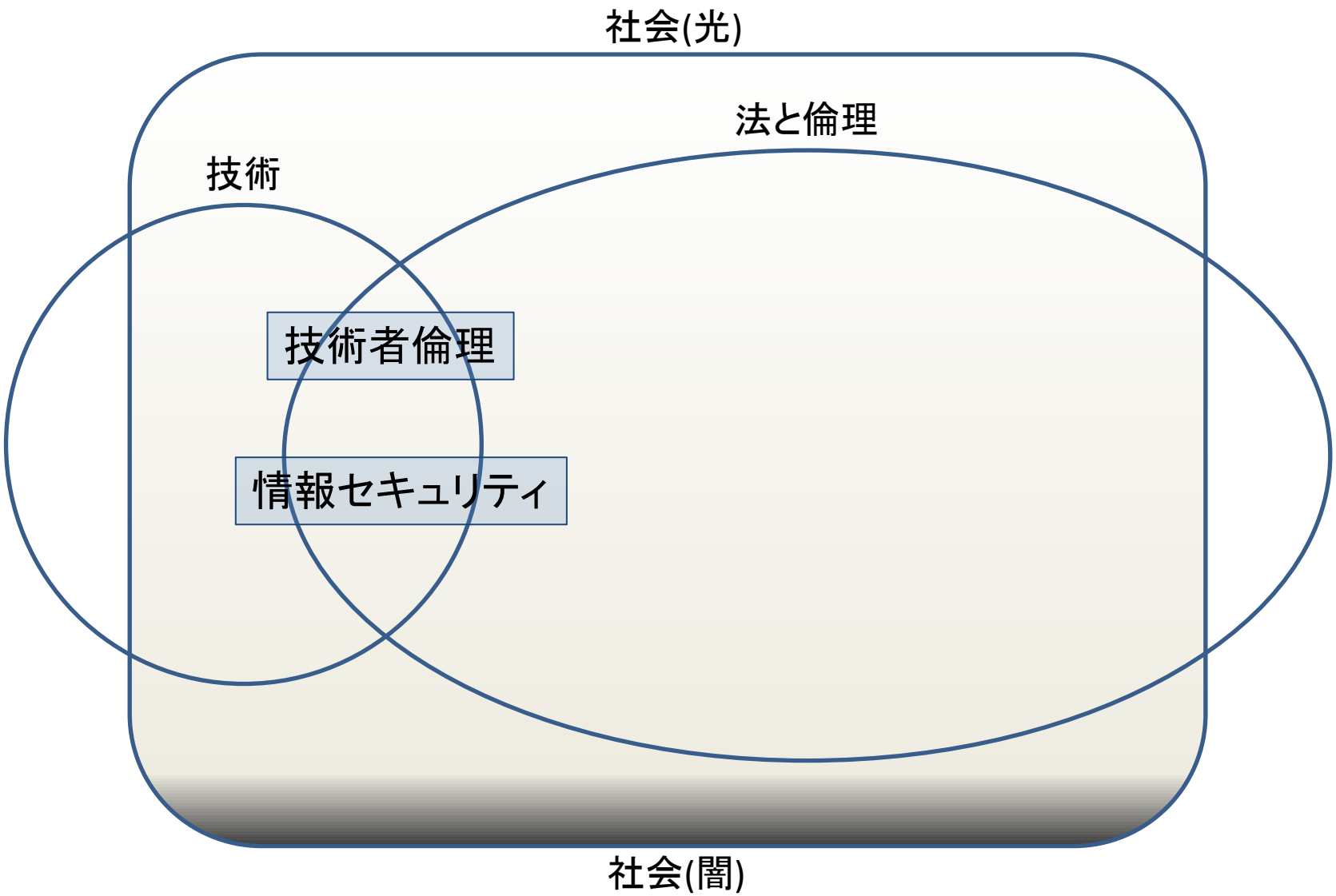
第3回　コンピュータの信頼性

2022/10/14

本科目でカバーする内容

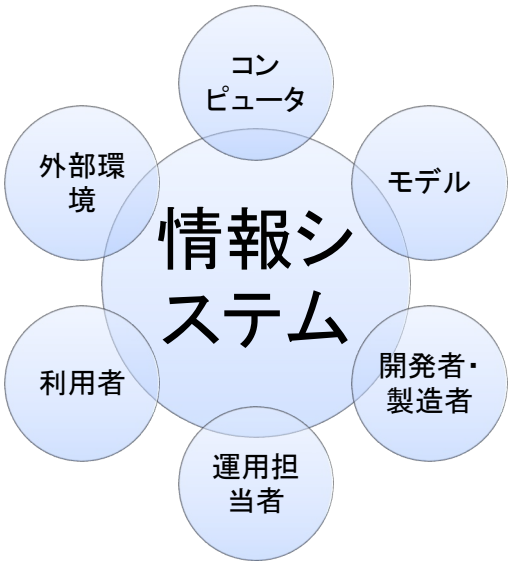


今日の内容



コンピュータの信頼性

- コンピュータ(情報システム)は信頼できるか？
  - コンピュータは正常に動作するか？
  - システムを作っている人は信頼できるか？
  - システムを動かしている人は信頼できるか？
  - システムを使っている人は信頼できるか？



ミニレポート

- 自動運転自動車の信頼性を高めるためには技術者として何をすればよいか、今日の講義で触れた事例・用語・概念を使って説明せよ。
- 提出先：Moodle
- 〆切：10/21(金)　16:10

コンピュータの計算結果は正しいか？

- 正しくない結果が出る要因
  - 表現誤差:10進数の0.1は2進数だと循環小数
  - 計算誤差:丸め誤差
  - ハードウェア故障(永久故障）:断線,絶縁破壊
  - ハードウェア故障(一時故障）:ノイズ,放射線
  - プログラムのバグ(ミス)
  - ハードウェアのバグ

計算ソフトの事例(2007)

- Excel 2007にバグ、誤った計算結果を*表示*
  - 850 * 77.1 = 100000 (正しくは65535)
  - 5.1*12850 = 100000 (正しくは65535)
  - 内部数値は正しいが表示ルーチンにバグ

- Microsoft社は修正パッチをリリース

出典　http://www.itmedia.co.jp/news/articles/0709/27/news018.html

Intel製CPUの演算バグ(1994)

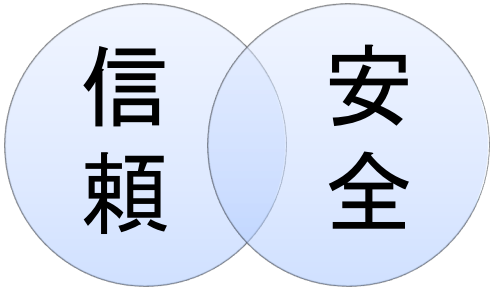
- Pentiumの浮動小数点除算に誤差
  - ex. 4195835.0／3145727.0 = 1.33374
    - 正しくは1.33382 (0.006%の誤差)
  - 高速除算手法の誤差補正処理に一部ミス
- 当時の反応
  - 「実際の影響はそれほど大きくない」(?)
  - 「当方の見積では無視できない影響がある」(?)
  - Intelは希望者のCPUを交換

CPU黒歴史 大損失と貴重な教訓を生んだPentiumのバグ
http://ascii.jp/elem/000/000/757/757002/

冗長性と自己検査機能

- 多数決論理
  - スペースシャトル：4台の独立コンピュータで相互検証、多数決論理、最悪時は5台目が登場
- 動作記録の保存
  - ATM処理 → 監査にも
- エラー検査ツールの利用
  - AT&Tでは計算能力の半分はエラー検査

信頼と安全



- 「信頼」
  - ある人や物を高く評価して、すべてを任せられるという*気持ちを抱くこと*。
- 「安全」
  - 危害または損傷・損害を受ける*恐れのないこと*。危険がなく安心なさま。
- 安全＝信頼か？

「大辞林（第三版）」より引用

  - 「安全だが信頼できない」
  - 「信頼て安全」「信頼できる(が安全でない)」

車の信頼性

- Q:中古車と新車、信頼性が高いのはどっち？
  - 車種は同じとする

ルンバの信頼性は？

- 信頼性の観点

1. 所定の機能が保たれる
   - ex. 部品が壊れない(故障しない)
2. 効果・影響が水準を満たす
   - ex. ゴミが除去される

「さすがルンバ」

「これ売っていいの？」

画像 https://www.irobot-jp.com/product/900series/index.html

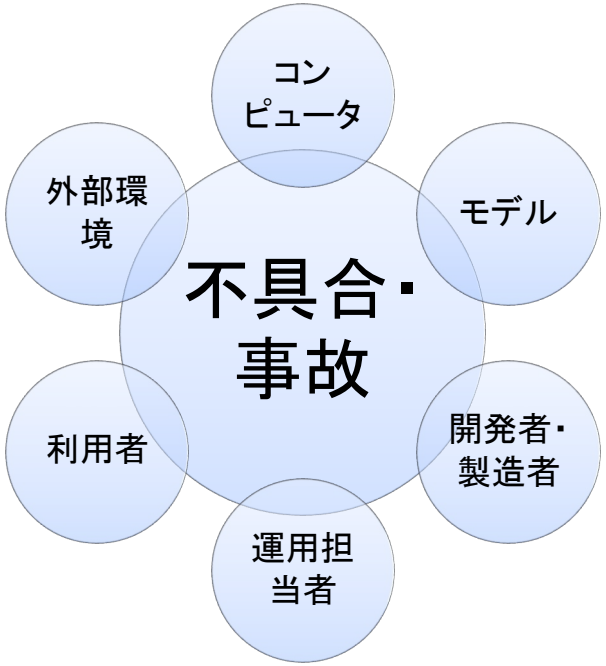
画像 https://twitter.com/yasubusin/status/552050563882287104

有名な事故･不具合の事例

- 航空宇宙分野
  - アリアン5ロケット1号機
  - エアバスA320
  - Mars Climate Orbitor
- 医療分野
  - Therac-25(放射線過剰放射事故)
- 金融分野
  - 証券取引所の停止
  - ジェイコム株大量誤発注事件(2005)

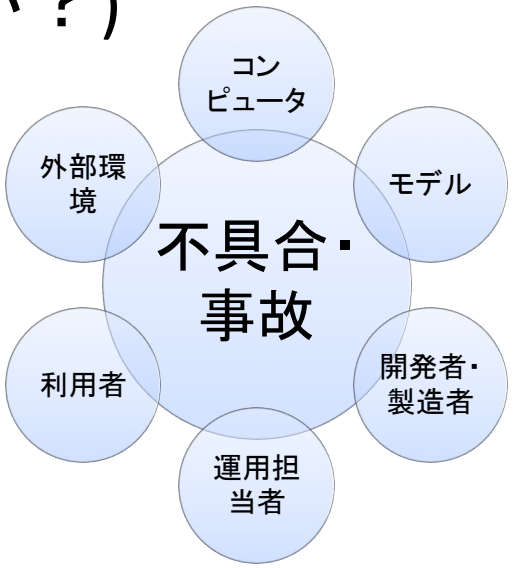
何が不具合を起こすのか？

- 複数の要因が絡む
  - 設計ミス、ずさんな実装、
  - 未熟で不注意なユーザ、
  - 劣悪なユーザインタフェース、
  - 等々



コンピュータ処理の大原則

- GIGO (Garbage-In Garbage-Out)
  - 入力が間違っていれば処理結果は正しくない
- 複雑系
  - 少しの違いが大違い(バタフライ効果)
  - 境界の決定が困難(何が影響するか？)



宇宙分野における再利用は安全か？

- ソフトウェア科学曰く
  - 「よくテストされた部品を再利用することでソフトウェアの信頼性を高める」
- アリアンロケット5の事例(1996)
  - 初打ち上げの40秒後に予定軌道を逸脱
    - 安全装置により自爆
  - アリアンロケット4号の制御ソフトを一部再利用
    - ロケット性能向上により、速度計算がオーバーフロー

Mars Climate Orbitor(1999)

- 衛星本体と地上システムで単位系の不一致
  - ニュートン・秒(衛星側)
  - ポンド重・秒(地上側)
  - 1ポンド重 = 4.45ニュートン
- 衛星の軌道補正に誤差が累積
  - 火星周回軌道投入に失敗
    - 予定軌道：高度140-150 km
    - 実際の軌道：高度57kmの軌道
- 地上側ソフトは過去ミッションから再利用

再利用の注意

- 元の利用範囲を超える場合は再テストを！

– ex.民生品の宇宙利用

事前テストすれば分かるが、事前テスト自体が困難なことも

GIGOシリーズ

- 請求ミス
- 誤認
- 消えた年金問題
- 学習データの汚染

請求書の間違い

- 比較的単純
  - 電気代63ドルが630万ドルに
    - 新米担当者の入力ミス
  - 101歳の老人の自動車保険のかけ率が十代の若者と同じに
    - 100歳までしか想定していなかったため
- プログラムの設計ミス、実装ミス、理解不足
  - 妥当性チェック、確認処理、検証で回避

データベースの更新漏れ

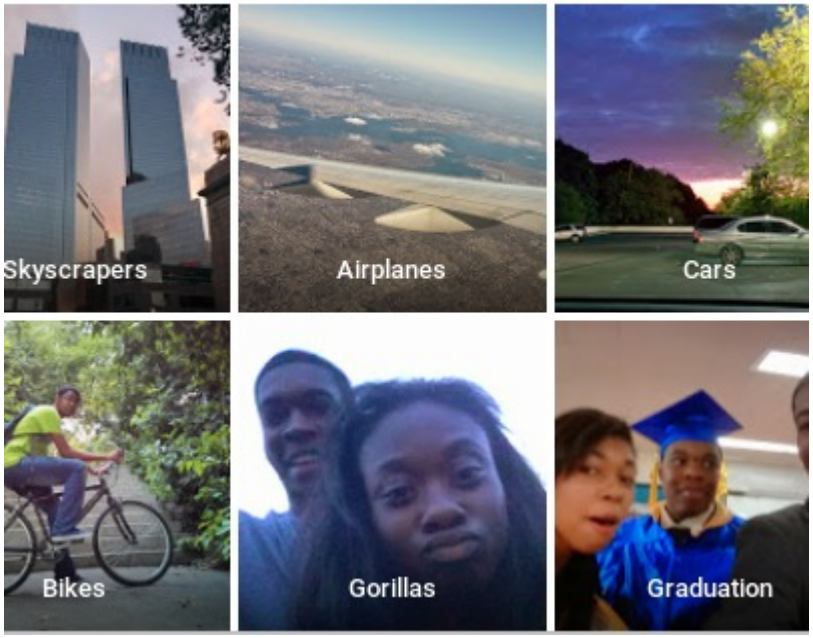
- 納税したのに未納扱いのまま
- データベース入力ミスで別人として登録
  - 行政サービスを受けられない
  - 犯罪歴のある別人と一致
- 引っ越してきたら性犯罪加害者扱いされた
  - 加害者住所データベースの更新漏れ

消えた年金記録問題(2007)

- 納付者を特定できない年金納付記録
  - 厚生年金：4000万件,国民年金:1000万件
  - 基礎年金番号に統合できない
    - 年金受給額の減額、受給資格の喪失
  - 名寄せ(氏名、性別、生年月日)も怪しい
    - 紙台帳を電子台帳にする際の入力ミス
- 年金保険料の納付記録が残っていないケース
  - 昔は集金係に現金で渡していた→横領(?)
  - 領収書などで納付事実を証明できた人はセーフ

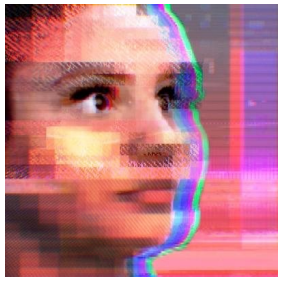
Googleのゴリラ問題(2015)

- 写真の自動判別機能で起きた問題
  - 黒人２人が映った写真を「Gorillas」と判別
  - Googleは謝罪。根本解決できず(2018)



出典　https://twitter.com/jackyalcine/status/615329515909156865

MicrosoftのチャットAI (2016)



- チャットボットTay
  - https://twitter.com/tayandyou
- 人間との対話データを元に学習・応答
  - サービス公開後16時間で公開停止
  - ヘイト・差別発言を学習してしまった

自動判断シリーズ

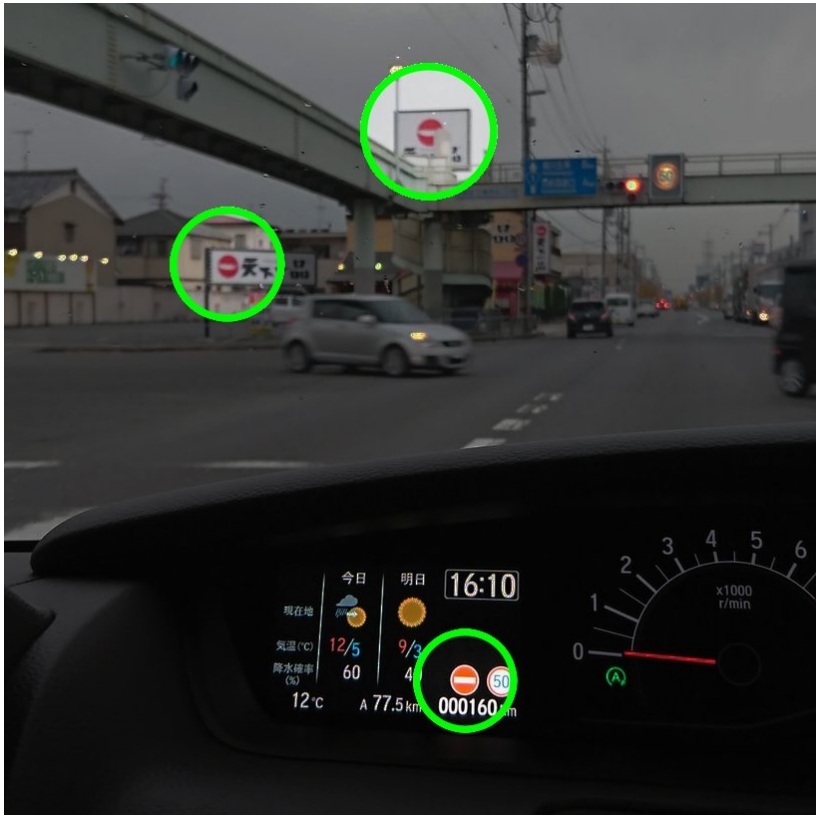
- 飛行機
- 自動車
- 試験

エアバス A320の事例

- 1988年～1993年に4機のA320が墜落
- A320の特徴
  - 全面的なフライバイワイヤ(自動操縦機能)を搭載
  - コンピュータによる制御優先の弊害
    - システムに不慣れなパイロットによる下降率の数値入力ミス
    - システムが着地を検知できず、パイロットの強制ブレーキ操作を妨害
- 自動制御から手動制御への簡単な切り替え機能が必須

人とコンピュータは異なる

- 「天下一品」は「進入禁止」？



出典　https://twitter.com/_gyochan_/status/938240168078622720/photo/1

自動運転走行車の死亡事故(2016)

- 信号のない交差点で白いトレーラートラックに衝突
  - Tesla社のModel Xがオートモードで走行中
  - 運転手は車からの警告を無視して手放し運転
- なぜ巨大なトレーラーの検知に失敗した？
  - Teslaは画像認識方式。他社はレーザー/レーダー

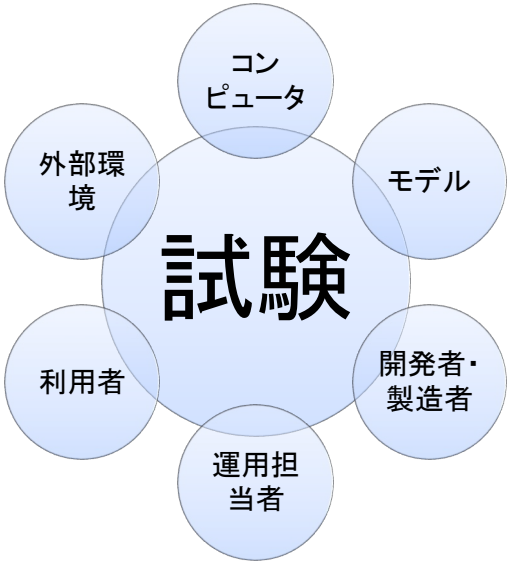
画像　https://www.flickr.com/photos/ntsb/35366284636/in/dateposted/

自動運転システムの指針

- システムの振る舞いを常時認識可能なこと
  - 加速中?減速中?方向転換中?
  - 「いつでも人間が運転をひきつげる」こと
- 経験あるユーザの期待通りに振る舞うこと
  - 「人間では普通やらない」ような動きをしない
  - 「システムが」人間に合わせる
- 仕事量が少なすぎると危険
  - 注意散漫になり緊急事態に対応できない

eテスティング

- ハイステークス・アセスメント
  - 入試・人事(採用・昇進)など
  - *結果が受検者又は組織にとって重要であるため，信頼性及び妥当性が高くなければならないアセスメント。(JIS X 7221 :2011)*
- 出題・採点の公平性は？
  - 項目反応理論の利用
    - SPI,TOEIC等



試験誤判定の事例(1999)

- CTB/McGrow-Hill社製標準試験の採点誤り
  - 毎年900万人の生徒が受験する学力試験
  - 「読解力」「数学」の採点プログラムにミス
    - 成績が突然下がった
  - ミス発見後も当初は学校関係者に告知せず
- ニューヨーク州の数千人の生徒に影響
  - 9000人が夏季講習送りになった
  - 解雇された学校長・教師も
- 意思決定を全面的に自動化する危険性

システムの欠陥

- 通信システム
- 在庫管理システム
- 空港システム

AT&T社のシステムダウン

- 1990年1月
  全国規模の9時間の通話･データ通信途絶
  - 5000万回線以上の通信が途絶
  - 400万行に及ぶプログラムに含まれていたエラー
  - 厳密なテストを行うも特異な事象の組み合わせを予測できなかった
- 1991年6,7月
  東海岸と西海岸の主要都市で電話網がダウン
  - 200万行に及ぶプログラムのわずか3行の変更（タイプミスを含んでいた）
  - 変更後なぜかテストせず

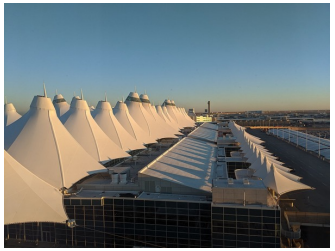
Warehouse Managerの事例

- 在庫管理システム　Warehouse Manager
  - 商品の注文ができなくなった
    - 処理に非常に時間がかかるため顧客の長い行列ができた
  - 価格や在庫量について誤った情報が提示された
    - 価格が低くければ会社の損失、高ければ顧客が逃げる
  - 2人が同時に操作を行うと装置がロック状態になった
  - 会計情報が消され、税務報告ができなくなった

トラブルの源泉

- Warehouse Managerのトラブルの源泉
  - 技術的な困難さ
    - あるOSから別のOSへソフトウェアを移植
      その時点で不具合を把握
  - 間違った管理法
    - 検査が不十分
- 不誠実さと対応の悪さ
  - 移植前の実績を顧客に説明
  - 発生した不具合を特異なケースと説明

立ち往生した空港システム



画像出典

- デンバー空港手荷物処理システム
  53平方マイル（135平方km）
  - 想定:手荷物を自動で空港各所に10分で配送
    → 搬送車の衝突、誤配送、荷物の落下など
  - 問題点
    - 搬送車のスキャナの汚れや調整不足による誤検知
    - 電源システムの容量不足
    - ソフトウェアエラー
  - 解決までに4回延期(1993年10月→1995年2月)

システムの運用が大幅に遅れた原因

- システム開発や検査時間が不十分
  - 開発2年＋検査6週間
    - 同規模のフランクフルト空港では開発6年＋検査2年
  - プロジェクト開始後に大幅な仕様変更
    - 当初United Airlines向け → 空港全体に拡張（14倍の規模）

発注側にも責任がある

稼働検査

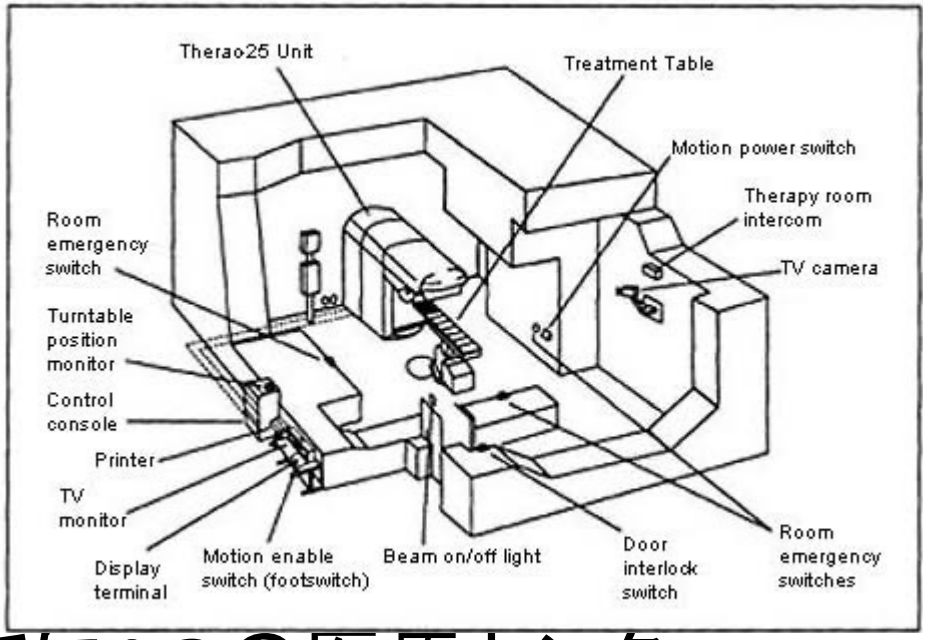
- 稼働検査は大変重要
  - しかし、比較的切り詰められやすい
    - 納期や経費の都合
- 独立実検証
  （Independent verification and validation）
  - 開発者、顧客とも異なる第3者が実施
  - “敵対者”として欠陥を探し求める
  - コストは高い vs. 安全性向上

Therac-25の事例

放射線過剰照射事故

http://www.computingcases.org/case_materials/therac/analysis/SocioTechnical_Analysis.html

- Therac-25
  癌患者に対するソフトウェア制御による放射線治療装置
  - 電子線ビームとX線ビームの使い分け
  - ビームに応じた安全装置の交換
  - コンピュータ制御されたターンテーブル



- 1985年から1987年までに4つの医療センターで6名の患者が過剰被爆の被害にあう

Therac-25 : 設計ミス

- 先行機との違いは完全コンピュータ制御
  - 先行機 Therac-6, Therac-20
    - 別ハードによる防護機能（安全装置）付
  - Therac-25では安全装置は不要とされた
    - 流用したTherac-20のソフトウェアが正常動作しているものとした
      → 実はバクがあり、Therac-20は防護機能があったから事故に至らなかっただけ
- 頻繁に誤動作
  - たいてい照射量不足という誤り
  - 操作技師がエラーメッセージになれてしまった
  - エラーメッセージは番号のみで説明無し

Therac-25：ソフトウェアのバグ

- 機器の使用前にチェックプログラムが走るが、異常が見つかった場合でも再チェックを行わず治療が開始されることがある
- 操作技師の機器調整の修正指示を受け付けない部分があった

Therac-25：事故原因多発理由

- 第1～4の事故
  - 未経験（大量被爆を見たことがなかった）
  - いくつかの問題点を発見し修正
    - 安全性が5桁上がったと宣言（真の原因は不明のまま）
  - 高価な装置の撤去のためらい
    - 治療機械の損失、病院収入の減少
    - 治療の損傷原因が装置によるものとは未立証
    - 原因が明確になっても製造会社が保証するかはっきりしていない
- 第5の事故で欠陥を認める
  - 1年がかりで改良
    - ハードウェアによる防護装置も導入

Therac-25：過信

- 安全性に過信
  - 患者が「焼けるようだ」と訴える
  - 操作者は「ありえない」と答える
- 致命的な過信
  - ハードウェアによる防護装置を削除したこと
    - 複数のチェック機構の重要性
  - 頻発するエラーの無視
    - 操作者の感覚の麻痺

Therac-25：結論と教訓

- 小さな設計ミスや実装ミスはつきもの
  - 利用者の不注意
  - 必要な経費を節約
  - ソフトウェア開発者の専門家としては、あるまじき仕事（不注意、手抜き）
  - 責任のがれ
- 教訓：重要な因子
  - 個々人の管理責任
  - よい訓練
  - 説明責任

リスク

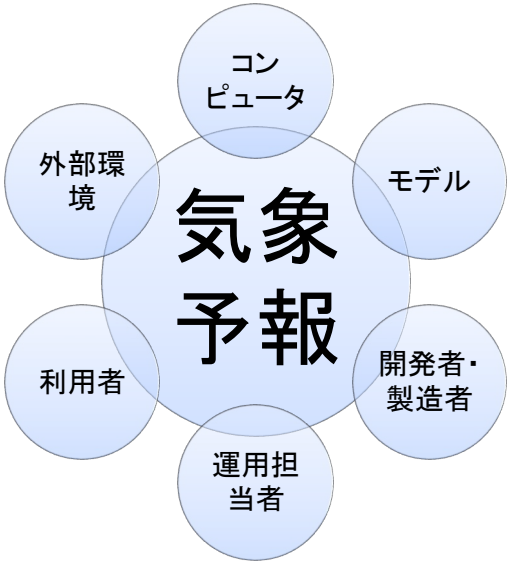
- 日々の生活は、技術への信頼に支えらえている
  - 例：エレベータに乗る
- 道具や技術が巨大化・複雑化 → 被害コストの増大
  - 例：散歩での他人との接触 vs. 車の接触事故
- 安全性は徐々に向上（例：車）
  - 運転者の責任についての教育向上
  - システム故障時の搭乗者保護
  - ITS（Intelligent Transport System :高度道路交通システム）
    - リスクを理解して合理的な予防策をとる
  - それでも絶対安全とは言えない

不確実性は悪か？

- 例：天気予報
  - 予報が外れることはある
  - 予報が外れた時、予報した側に責任はあるか？
  - 「予報が外れる」＝「コンピュータの誤り」か？
- 予報のジレンマ
  - 「間違った予報を出したせいで被害が出た」
  - 「予報を公表しなかったせいで被害が出た」
  - 「予報を公表したせいで被害が出た(パニック)」

不確実性と社会的合意

- 「外れても良い」のはどういう場合か
  - 品質について社会的な合意があるか？
- 気象予報士制度(1993年気象業務法改正)
  - *予報業務のうち現象の予想については、気象予報士に行わせなければならない*
    - 昔は気象庁の独占業務
    - 解説者は資格不要



気候変動予測

- 自然現象のモデル化はどうやる？
  - 太陽活動の変動は？
- シミュレーションの入力データはどうする？
- 生物の影響は考慮する？
  - メタン発生要因の24%は家畜のゲップ
- 文明発達の影響は考慮する？
  - 50年後には世界中で電気自動車が普及？
- 外れた時の責任は？

アンケート

- Q:検索エンジンの出力は信用できないから、利用を禁止するべきである？
- Q:病院の投薬ミスはどの程度許容できるか？

エアバッグの信頼性

- HONDA エアバッグ・システム(1987)
  - 信頼度：99.9999%　で設計
    - 故障率：百万分の1
  - 部品単位のトレーサビリティ
    - どの車にどの部品を取り付けたか追跡可能

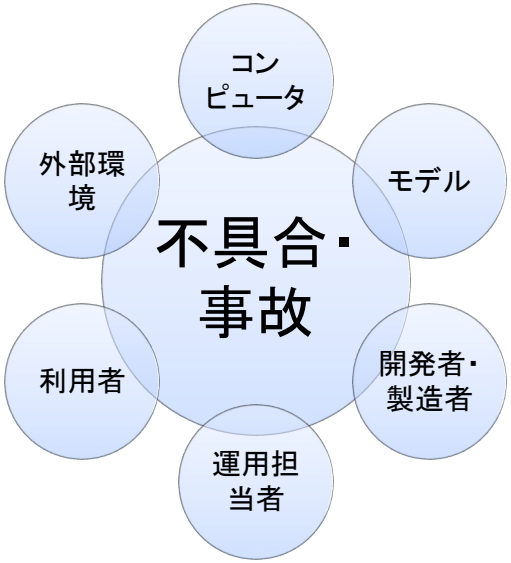
出典 https://www.honda.co.jp/50years-history/challenge/1987theairbagsystem/page04.html

アンケート

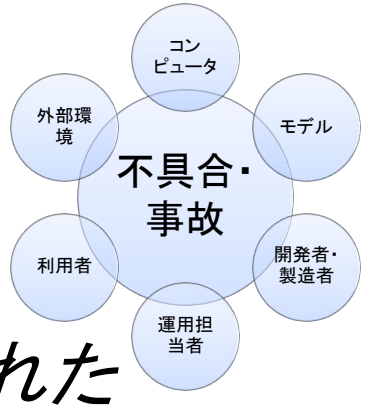
- Q: 販売される製品に不具合はあるべきではない。ソフトウェアも同じであるべきか？
- Q:Windows OSを利用することは社会的なメリットであるから、不具合の存在は許容すべきである？
- Q:ソフトウェア開発は資格制にするべきである？

コンピュータを使うべきか

- 使うことによる信頼性の向上
  - 人手でも「誤り0」はありえない
- 使うことによるリスク
  - 使わないことによるリスク
- 開発者・専門家の責任
  - 「誤り0は困難」≠「事故が起きても良い」
  - リスクの分析、バックアップシステムの計画



複雑システムはなぜ事故を起こす？



- How Complex Systems Fail(1998)
  - 事故の発生要因を18項目に整理
  - *医療システムの事故防止のために書かれた*
- 抜粋
  - 複雑システムは故障を内包したまま動作する
  - 人の介在は事故を招き、また、事故を防ぐ
  - システムの変更(改善)は未知の不具合を招く
  - 安全性はシステム要素ではなく全体に宿る

ユーザインタフェースと人的要因

- よい設計のユーザインタフェースは多くの問題を解決しうる
- エラーを防ぐ
- 安全なシステムの構築には
  - 例：自動飛行システム
    - 自動システムが行っていることを明示する
      - 目的高度進入時の上昇率の変化など
    - システムはパイロットの期待する通り動作
      - 手動操作が完了したら自動運転に自動で戻すなど
    - 仕事量が少なすぎると危険
      - 退屈、怠慢の原因

コンピュータシステムの信頼性・安全性(1/2)

- コンピュータは間違えない
  →（ある意味で）正しい
- コンピュータは不具合を起こさない
  → 正しくない

- 大抵のコンピュータシステムは、複雑すぎて誤りを含まないプログラムを作り出すことは不可能に近い
  - 人間の限界？

コンピュータシステムの信頼性・安全性(2/2)

- 社会は、故障した時の被害を覚悟しているか？
  - スマートスピーカー「Amazon エコー」でニュースの声を注文と勘違いし大量の誤発注（2016）
    → Amazonはトラブルで誤発注された商品のキャンセル、または払い戻しを対応
  - ATMの故障、電話・電力網の故障・・・
- コンピュータシステムの効率性を重視し、依存する代わりに安全性を犠牲にしているか？
  - 医療システム、航空機システム・・・

教訓（1/2）

- 開発過程全般で良質のソフトウェア開発技法を使用しなければならない
- たいていの事故は未知の科学的原則から起こるのではなく、よく知っている標準的技術の適用ミスから起こる
- 事故というものは技術的な改良だけで回避できるものではなく、システム全体の開発や運用を制御する必要がある

教訓（2/2）

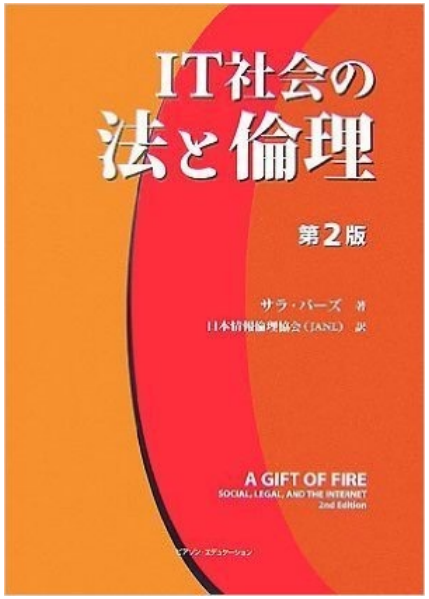
- コンピュータ以前のシステムでのハードウェアによる安全機構は特に重要
  - ソフトウェアによる制御の安全性の確証があるまで外すべきではない

むすび

- コンピュータの信頼性にかかわる多くの事項は、その他の技術に関連して以前から問題になっていたこと
- 複雑さが「誤り」、「見落とし」、その他の不確実さの原因
- 新技術には「学習曲線」がある
  失敗から学ぶことで失敗の「頻度」を軽減できる
- 道具を利用することにより得られるリスクと利点を比較すべき

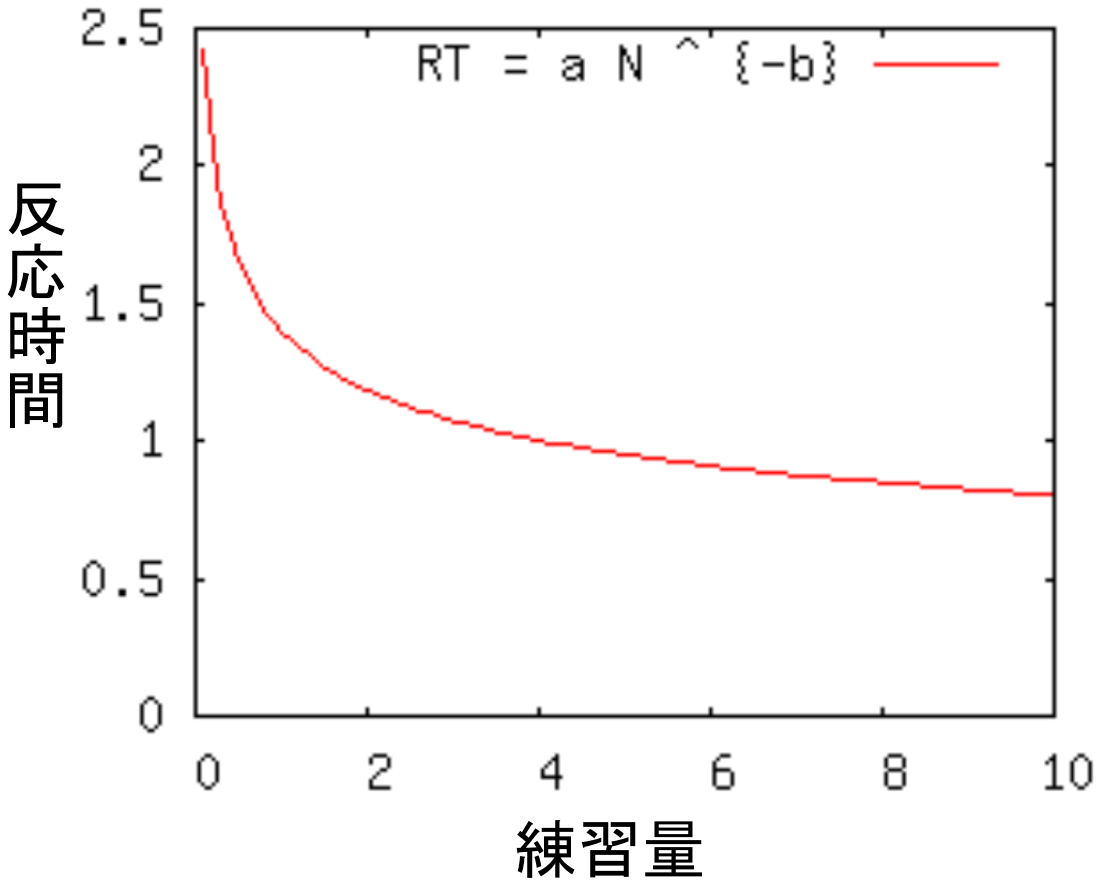
参考文献

- IT社会の法と倫理



ピアソンエデュケーション
ISBN:978-4894714304

(参考)学習曲線



学習曲線

RT = aN^{-b}

N:練習量, RT:反応時間

a=1.40, b=0.24

(ピロリとアンダーソンによる実測値)