ソフト不具合によるシステム障害の軽減方策

2013年3月23日（土）

航空運航システム研究会（TFOS.SG)
航空システム部会　松田　宏

問題提起と対策提案の視点

- 航空システムのコンピュータ依存度が高まるにつれ、ソフト不具合によるシステム障害のリスクが高まっている。
- 航空以外のシステム障害事例でも、類似の障害が航空システムで起こる可能性があるため、参考になるはず。
- システムの信頼性を大きく左右するソフトの品質は、次のすべての段階の品質管理の積み重ねの結果である。
  - システム分析（要求仕様の定義）
  - ソフトの設計（概念設計、基本設計、詳細設計）
  - ソフトの製作と試験（プログラム・モジュール、結合、総合）
  - システム・インテグレーションと試験、設置調整
  - システムの運用保守や操作
- ソフトの単純なバグも困るが、上流工程に問題が多い。

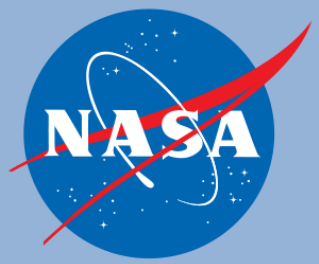
事例1：ハイフンひとつでロケット爆発？

- 1962年7月に打上げられたNASAの金星探査衛星を搭載したマリナー1号は打上直後に軌道をそれ、指令破壊により爆破された。
- ロケット設計者は速度測定レーダーのデータをスムージング処理して使うようRに¯(bar)がついた手書きの計算式をプログラマに渡した。
- しかし制御プログラムは”¯”なしで記述されたため、速度データが正しく処理されず、ロケットは予定の軌道からそれた。
- 後年、この話が「“－”（ハイフン）のミス」と誤って伝えられ、ソフト業界の伝説になった。

マリナー1号の打上
（293秒後に爆破）

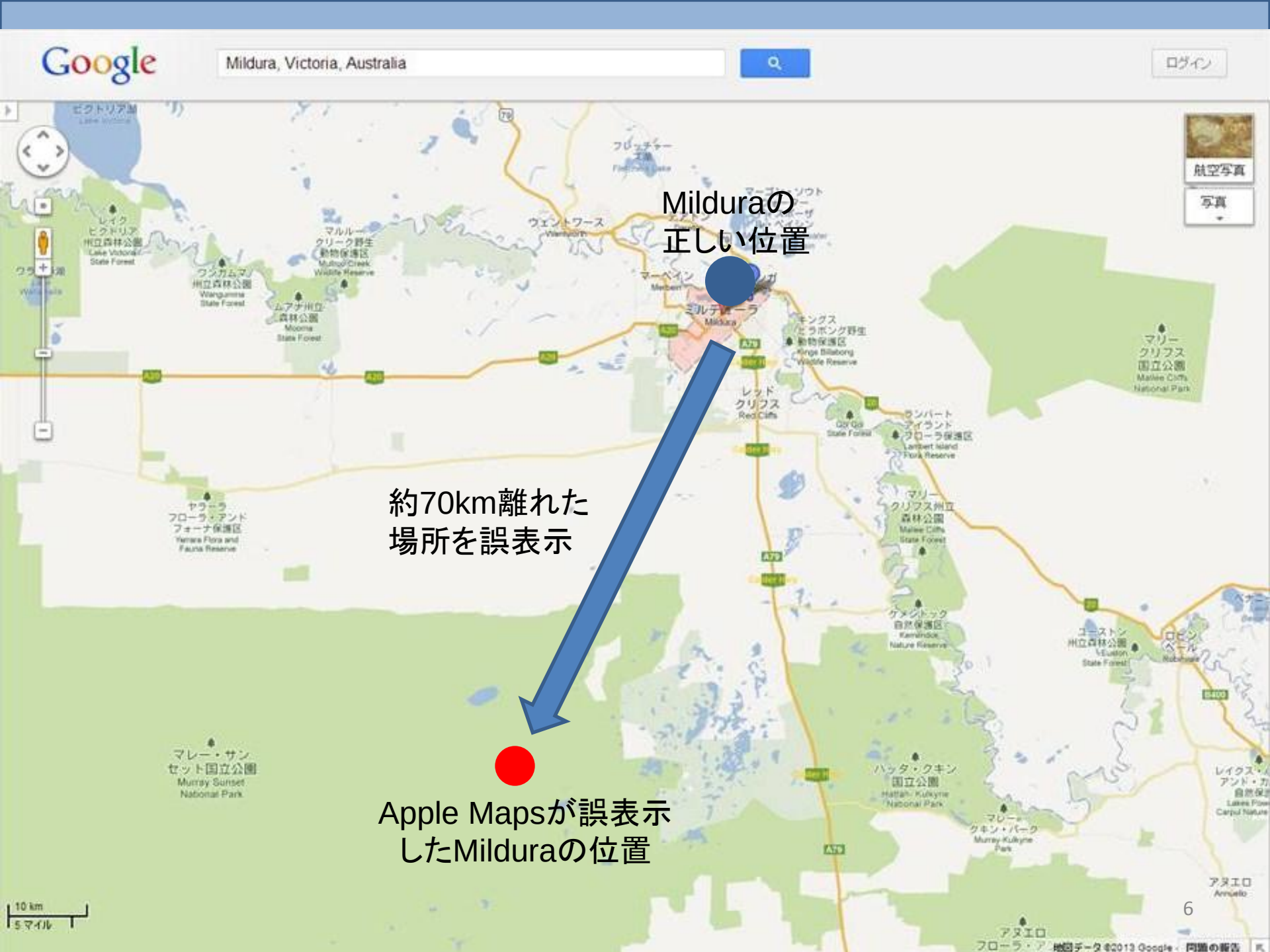
解説：要求側と設計者との常識の違い

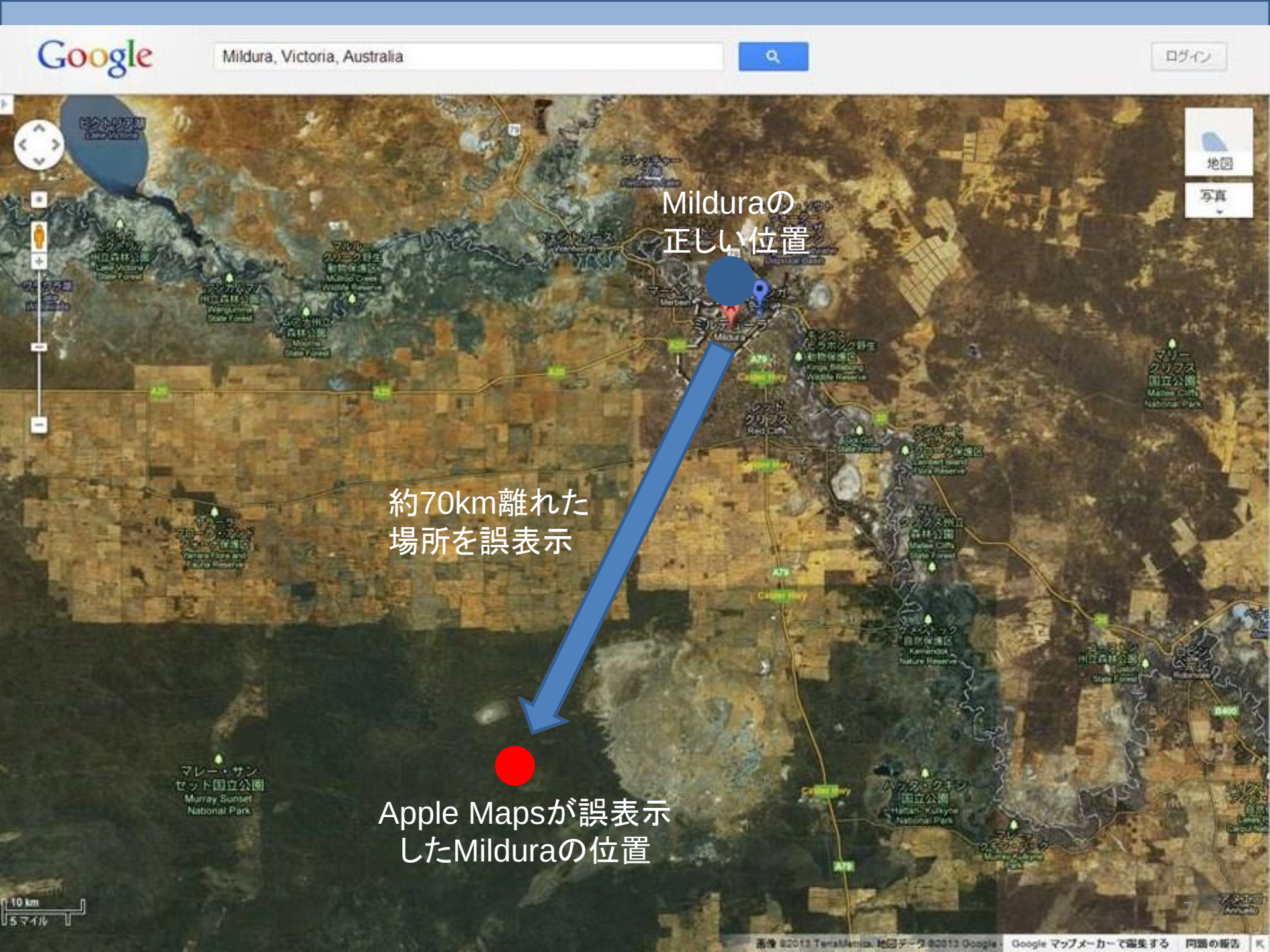
- ロケット設計者（科学者）は、十分な情報を提供したと信じていたはず。測定データのスムージングは常識なのだから。
- プログラマ（技術者）は、スムージングという概念を知らず、計算式の意味を誤解したと思われる。
- プログラマは自分の解釈でプログラム（FORTRAN）を書き、試験し、結果が彼が考える意味で正しいことを確認したはず。
- 打上直後のデータでは誤差が小さくて、プログラムの間違いがわからなかった可能性はある。
- 依頼者と作成者の間に常識の違いによる誤解があると、お互いに確認したつもりでも認識の差がわかりにくい。



事例2：間違った地図情報で遭難

- 【2012年12月10日 AFP】
  オーストラリアの警察当局は、地図の間違いで死亡の危険があるとして、米アップルのiPhoneの新しい地図アプリApple Mapsを使用しないようドライバーに警告した。
- ビクトリア州警察によると、ここ数週間に同州内陸部の町ミルデューラ（Mildura）へ行こうとした車が「道を外れ」、約70キロずれたマレー・サンセット国立公園の真ん中へナビゲートされる事態が続出した。
- 救出されたドライバーはこの数週間で6人。うち何人かは食料も水もない状態で最長24時間さまよった。公園内は気温が摂氏46度に達することもあり、人命に関わるという。
- この事件でアップル社の幹部2名が辞職した。


Google
Mildura, Victoria, Australia
ログイン
Milduraの
正しい位置
約70km離れた
場所を誤表示
Apple Mapsが誤表示
したMilduraの位置
航空写真
写真
10 km
5 マイル


Google
Mildura, Victoria, Australia
ログイン
Milduraの
正しい位置
約70km離れた
場所を誤表示
Apple Mapsが誤表示
したMilduraの位置
地図
写真
10 km
5 マイル

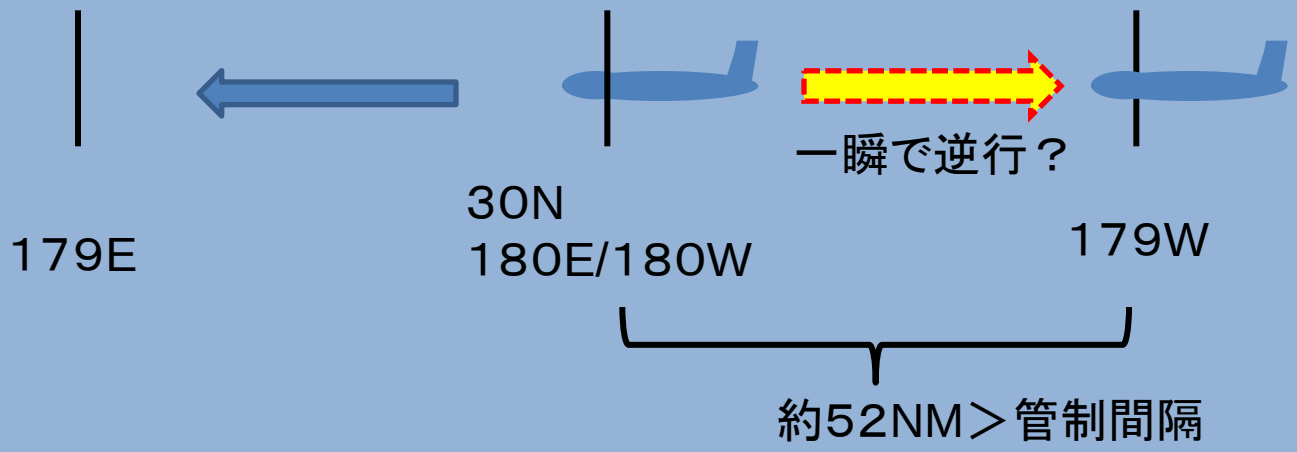
解説：データの間違いで事故が起きる

- カーナビは、自動車運転者が目的地までの道や渋滞／工事などの状況を知ることが目的で、メーカーも責任を認識？
- スマホの地図情報サービスは、目的があいまい。コスト削減を優先し、信頼性の低いデータを提供した可能性がある。

- 昔々、食べられるキノコと毒キノコの写真を間違って逆に掲載した百科事典を発売し、回収した出版社があった。
- 今では、あらゆる情報が瞬時に世界中に広がってしまう。人命や財産にかかわる重要情報が間違っていると大変だ。
  - 医療情報（医師向け／一般向け）
  - 政治／経済関連ニュース（誤報で国際紛争や金融パニックも）

事例3：FMSが現在位置を誤表示（1）

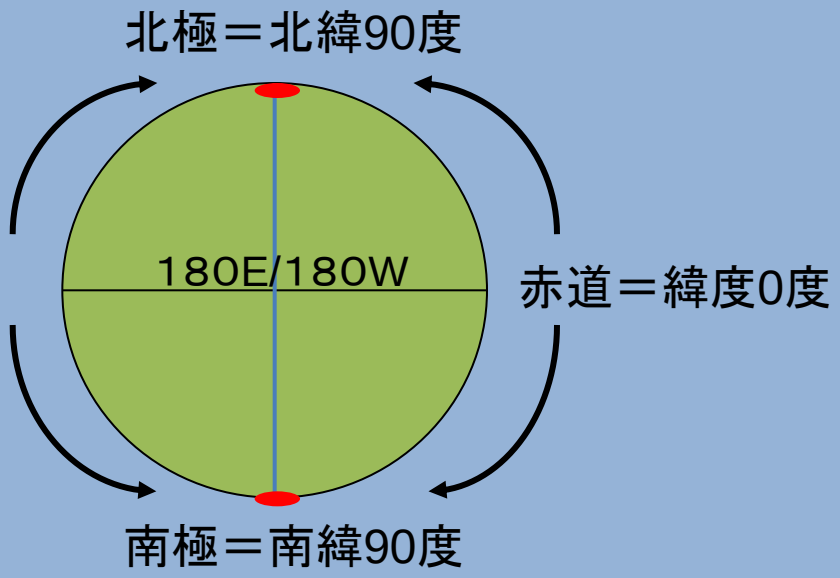
- 中部太平洋を西に飛行中の旅客機が180°に近づいたら、FMSが現在位置をW179° 00′ 00″と表示した。
- 位置計算の結果が179.9998度となったのを179度00分00秒と誤処理したためらしい。
- パイロットは表示が異常であることに気付いたため問題なかったが、管制間隔を考えると無視できない大きさの誤差。



事例3：FMSが現在位置を誤表示（2）

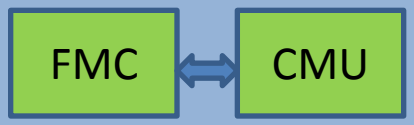
- 飛行中の旅客機が経度180度を通過した際、FMSが現在位置を“S180W180”と表示した。
- そんな場所は地球上にはない（ソフトウェア開発時のデフォルト値だった）ので、パイロットは異常に気がついた。

S180（南緯180度）
という場所は地球上
にはない！



事例4：FMSは計算中に反応できない

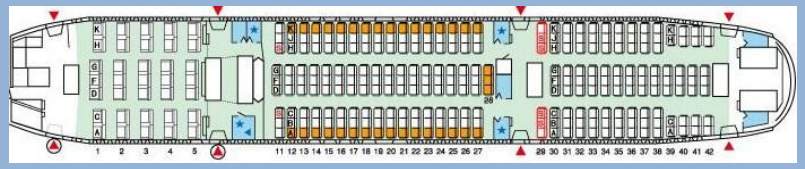
- 航空機が管制機関と情報交換するためのデータリンク・システムCPDLC（Controller Pilot Data Link Communication)がなかなかログオンできないことがある。
- 昔からある現象で、VHFとSATCOMの干渉などの可能性も指摘されているが、地上で再現できないため原因は不明。
- 可能性として、経路の変更等で飛行管理システム（FMS）が再計算をしているタイミングで通信管理装置（CMU）とのデータ交換が拒否されるためでは、との指摘がある。
- FMS搭載機に後からADSやCPDLCなどのFANS機能を追加した古い機種に多く、単体では正常なシステムが相互関係で問題を起こしている可能性がある。



解説：起こりがちなソフト不具合

- ソフト開発には「鬼門」のようなミスが多発する場所がある。古典的なもので経験豊富なベテランならそんなミスはしないが、若い人は同じ落とし穴に落ち込みがちである。
  - 大晦日の23時59分59秒～元旦の0時0分0秒の間
  - うるう年、うるう秒、古くは2000年問題、これからの2038年問題など
  - 赤道（北緯0度／南緯0度）の前後、東経180度／西経180の前後
- 部分的なハード障害や致命的でないファイル・オーバーフローで、エラー・メッセージが溜まってシステムがパンクする。
- 想定外のデータは排除するのが常識だが、そうした考慮は忘れがち。最悪の場合、異常停止の原因になる。
- 処理に時間がかかるのに、短時間でタイムアウトになる。

事例5：座席指定10万席分が消失



- 2012年11月、全日空国内線2013年2月分の座席指定約10万席分が、予約システムの不具合で取り消されていた。
- 対象は、11月26日午後6時までに購入した国内線航空券のうち2013年2月搭乗分。
- 航空券の予約そのものには影響がなく、事前の座席指定情報のみが取り消されていた。
- ホームページ、国内線予約・案内センター、特設のコールセンターで改めて座席指定が必要となった。
- 原因は、営業担当者の操作ミス。２人でダブルチェックする体制にもかかわらず、予約システムの時刻表情報を更新する際に手順を守らず、誤って予約情報を消去してしまった。

解説：手順を守れというだけでは不十分

- ミスを避けるため、２人でダブルチェックしながら所定の手順を守って操作しなさい、というだけでは不十分。
- システムが操作の妥当性をチェックし、このデータを本当に消してもいいのですか、と確認を求める程度はやっていたはずだが、習慣的にOKを押してしまう事故はなくならない。
- 重要なデータ（この場合は3か月先の座席指定情報）を容易に消すことができ、復元のためのバックアップ・データがなかったというのは設計上の考慮不足。
- 重要データが消えても、前日までのデータのバックアップと当日のジャーナルによって復元できるように設計しておけばよかった。つまり、これは広義のソフトの不具合である。

事例6：原因不明でも対策は可能

- 昔々、人工衛星の運用管制システムを開発し納入したが、その衛星は打上に失敗し、開発チームは解散した。
- １年後の再打上げが迫った頃、お客様からおかしな現象が起きるので見に来て欲しい、とお呼びだしがあった。
- 管制室から800m離れた送信所から遠隔操作で衛星制御用のコマンド送信のリハーサルをしていたら、進行波管の高電圧電源のON/OFFを表示するランプが「ウィンク」するという。
- その部分の開発者は別な仕事で遠隔地に出張中。仕方なしに調べたが、複雑な条件が重なっていて原因不明。しかし実害がないことと、起きている現象だけは詳しく判明した。
- お客様と相談し、その現象を検出したら状態を正常に戻す処理を追加するという「姑息な」対策を講じ、問題は解決した。

参考：技術試験衛星の運用管制システム

郵政省電波研究所鹿島支所にて(1978年）
（現・独立行政法人　情報通信研究所　鹿島宇宙技術センター）

番外事例1：制御ソフト以前の問題？

- 2012年9月29日、「おむつ」などに使われる高吸収性ポリマーを生産している日本触媒の姫路製造所で、アクリル酸の中間タンクが爆発炎上した。死亡1名、重軽傷36名。
- 13:20頃、白煙を確認。13:48頃、消防に通報。14:35にタンクが爆発炎上。22:36に鎮圧。翌日15:30に鎮火。
- 事故調査委は中間報告で、タンク内に温度計がなく、温度管理（遠隔監視等）も不備だったと説明。消防に提出した図面には温度計があるとの矛盾した証言もあり、信憑性に疑問。
- アクリル酸は一定以上に温度が上昇すると熱暴走を起こし、温度が急上昇して爆発炎上する。冷却では止められない。

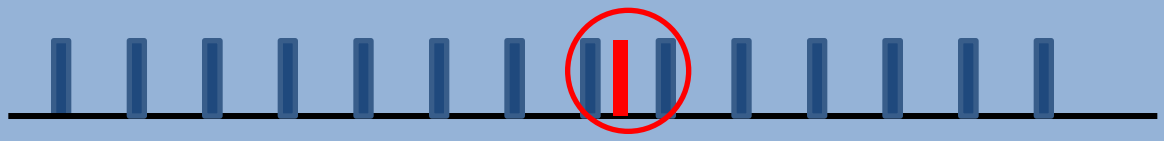
解説：化学プラントの制御

- 長期的な不景気のため古いプラントが稼働している例は少なくないが、1975年以降、デジタル計装制御システム（DCS：監視、フィードバック制御、シーケンス制御などをもつ計測制御用のデジタルシステム）は常識となっている。
- 近年、プラント各部の情報をリアルタイムで中央監視室に集中し、コンピュータによる自動監視・制御が進んでいる。
- また、プラントの自動監視・制御と上位の生産計画や操業計画などの機能の連携も進んでいる。
- 制御ソフトの開発や保守における生産性と信頼性を向上されるため、CASE(Computer Aided Software Engineering)の導入も進んでいる。事故を起こした工場も導入先のひとつ。

番外事例2：ソフト不具合と疑われて（1）

- 2010年に米国で、トヨタ製自動車の急発進事故が問題に。
- 原因のひとつは、アクセルペダルがカーペットにひっかかる現象で、リコールして改修した。
- もうひとつの原因として疑われたのが電子制御装置（ETCS）。社長が米議会で涙の弁明をしたが認められず、その間に米国車や韓国車がシェアを伸ばした。
- 2011年2月、米運輸省高速道路交通安全局（NHTSA)はトヨタ車の急発進問題に関し、電子スロットルに欠陥はないと発表。報告書はNASAの協力も得て10カ月かけてまとめたもの。
- ソフトの場合、不具合を証明するのも問題なしと証明するのも、専門家が長期間かかる高度で困難な仕事である。

番外事例3：ソフト不具合と疑われて（2）



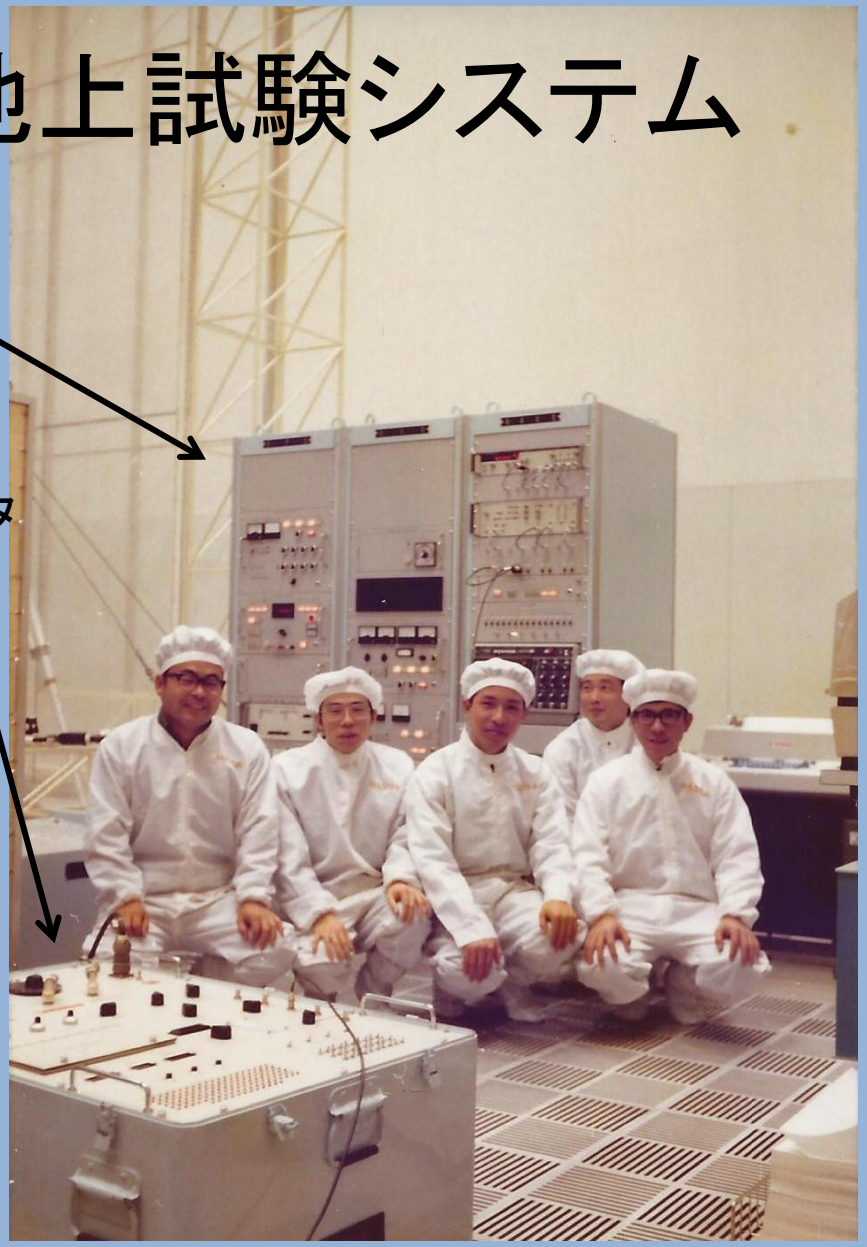
- 1970年代、国産人工衛星からのテレメータを受信して状態監視するシステムの開発中に、データの乱れが発見された。
- データは2秒ごとに繰り返す定型フォーマットだが、何故かごくまれにデータが乱れる。先ずはソフトが疑われた。
- ソフト側はいくら調べても問題はなく、受信データ中に余分なビットが混入している可能性が浮かんだ。
- 記録計で膨大なデータを収集して解析したが、隣の部屋の衛星から受信装置までは問題なし。しかし、コンピュータ側には余分な1ビットが明確に記録されていた。
- 最後に原因が判明。コンピュータの磁気ディスクが発生する強い電気パルスが衛星からのデータに混入していたのだ。

技術試験衛星の地上試験システム

テレメータ／コマンド送受信装置

データ解析装置（ミニコンピュータ）

人工衛星シミュレータ



宇宙開発事業団（NASDA：当時)筑波宇宙センターにて

ソフト不具合によるシステム障害

＜ソフト単独による障害＞

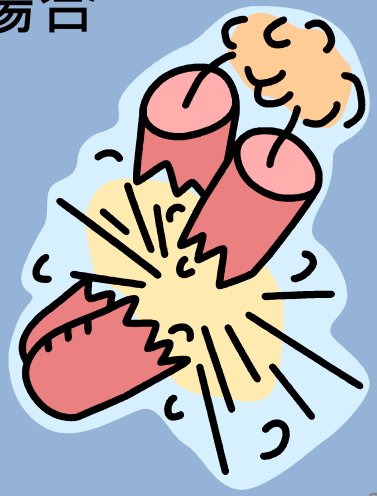
- システムが異常停止する。
- 正しいタイミングで処理されない。
- 入出力が適切に行われない。
- 処理結果が正しくない（数値計算の場合は誤差が大きい）。
- データが無くなる。化ける。

＜ハードが関連する障害＞

- ハード障害時の処理が適切に行われない。
  - 対応ソフトの不具合で異常停止
  - 予備系への切替えができず異常停止、など
- 容量オーバー時の対応が適切に行われない。

システム障害の影響度

- システム障害が事故に直結する場合
  - 人の身体、生命にかかわる事故
  - 物損など、経済的な負担をもたらす事故
- システム障害が事故につながる可能性のある場合
  - 代替システムや人間の関与で回避できる場合
  - 代替システムがなく、人間が関与しても回避が困難な場合
- 業務に支障があるが事故の可能性は少ない場合
  - 人間が代替して業務を継続できる場合
  - 人間では代替できず、業務が停止する場合
- 業務に大きな支障はないもの
  - 経済的な損失が発生する場合
  - 経済的な損失はない場合



対策1：ソフト不具合を減らす

＜開発以前の考慮＞

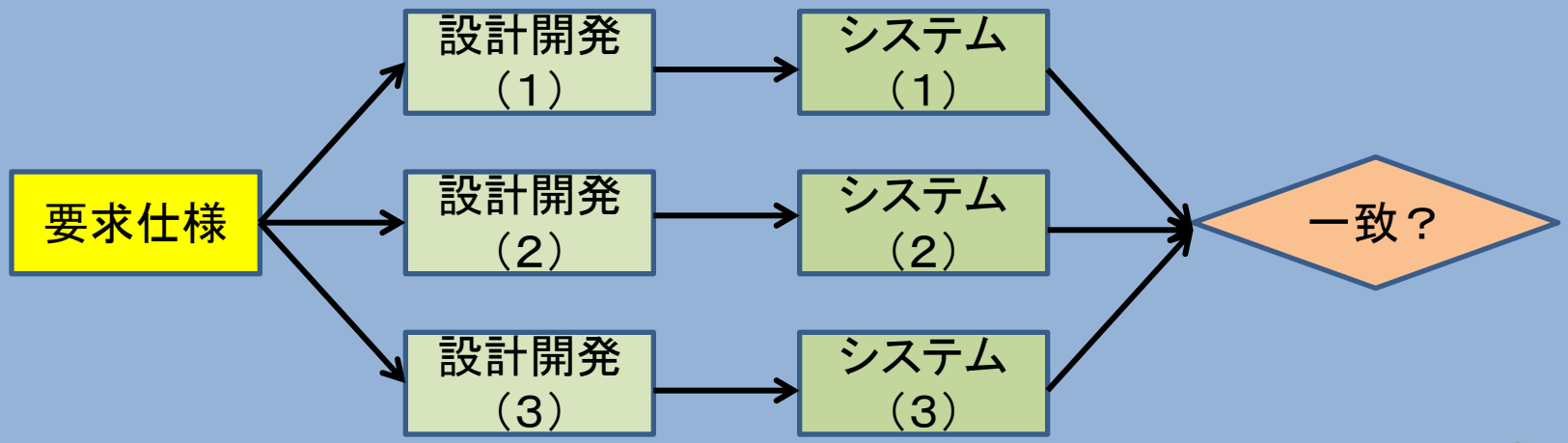
- システム以前に業務そのものを見直し、標準化しておく。
- 業務を熟知した利用者が関与し、要件定義に時間をかける。
- できるだけ機能を絞り込み、複雑なシステムにしない。
- 実績と利用経験を積み重ねながら、漸進的に改善改良。
  （画期的、革新的なシステムはリスクが高い）
- 起こり得る異常を広く想定し、対応の限界を明確にする。

＜開発時の考慮＞

- 適切な方法論とツールを活用する。
- 実績のある市販パッケージを活用する。

対策2：ソフトによるシステム障害を減らす

- システム構成機器の多重化は機器障害には有効だが、ソフトは同じなので、ソフト不具合の場合は意味がない。
- パッケージ・ソフトは多数の利用者が多様な環境で使用し、欠陥が早期に発見され、改修されるので信頼性が高い。
- 費用と時間はかかるが、同じ要求仕様に基づいて別々にソフトを開発し、結果の多数決を取る方法も提唱されている。



対策3:システム障害の影響を減らす

＜人間が関与する場合＞

- システム障害時に重要な機能だけは人間が対応できるように準備し、定期的に教育訓練を実施しておく。
- 同等ではなくても、重要な機能だけは代替できる別なシステムを準備しておき、障害が起きた場合はそちらで対応する。
- 障害の影響を受ける人達に自分達で備えをしてもらう。

＜人間が関与しない場合＞

- システム障害に備え別な対応手段を準備しておき、自動的に切り替えることにより最小限の機能を維持する。
- 機能を分散しておき、本システムに障害が起きた場合は個々のサブシステムが最小限の機能を自律的に継続する。

ご清聴ありがとうございました

＜次　回＞　いかがいたしましょうか？

- （今回の続き）
  - ソフト不具合の証明方法
  - ソフト原因を否定する証明方法
- （前回予告の内容）
  - 標準化と陳腐化の矛盾の解決方法
  - インターオペラビリティ（相互運用性基盤）の確保