宇宙機高信頼性ソフトウエア開発プロセスのアプローチ

2004年9月

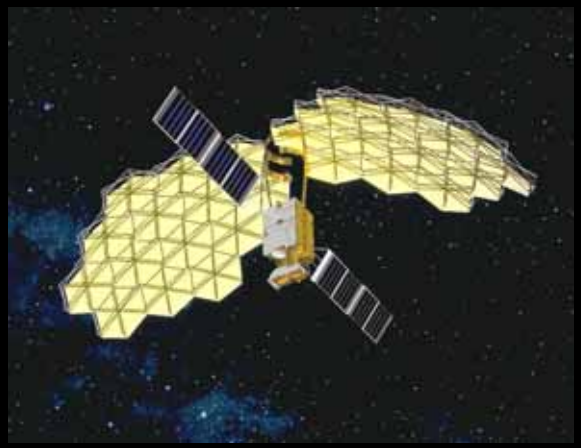
宇宙航空研究開発機構（JAXA)
情報技術開発共同センター
片平　真史、奥田　一実、石濱　直樹

目次

- 概要紹介
- 課題分析事例
- 設計解析
- 開発・検証
- まとめ・今後

宇宙用のソフトウエアとは….

宇宙用高信頼性ソフトウエアの種類と特徴



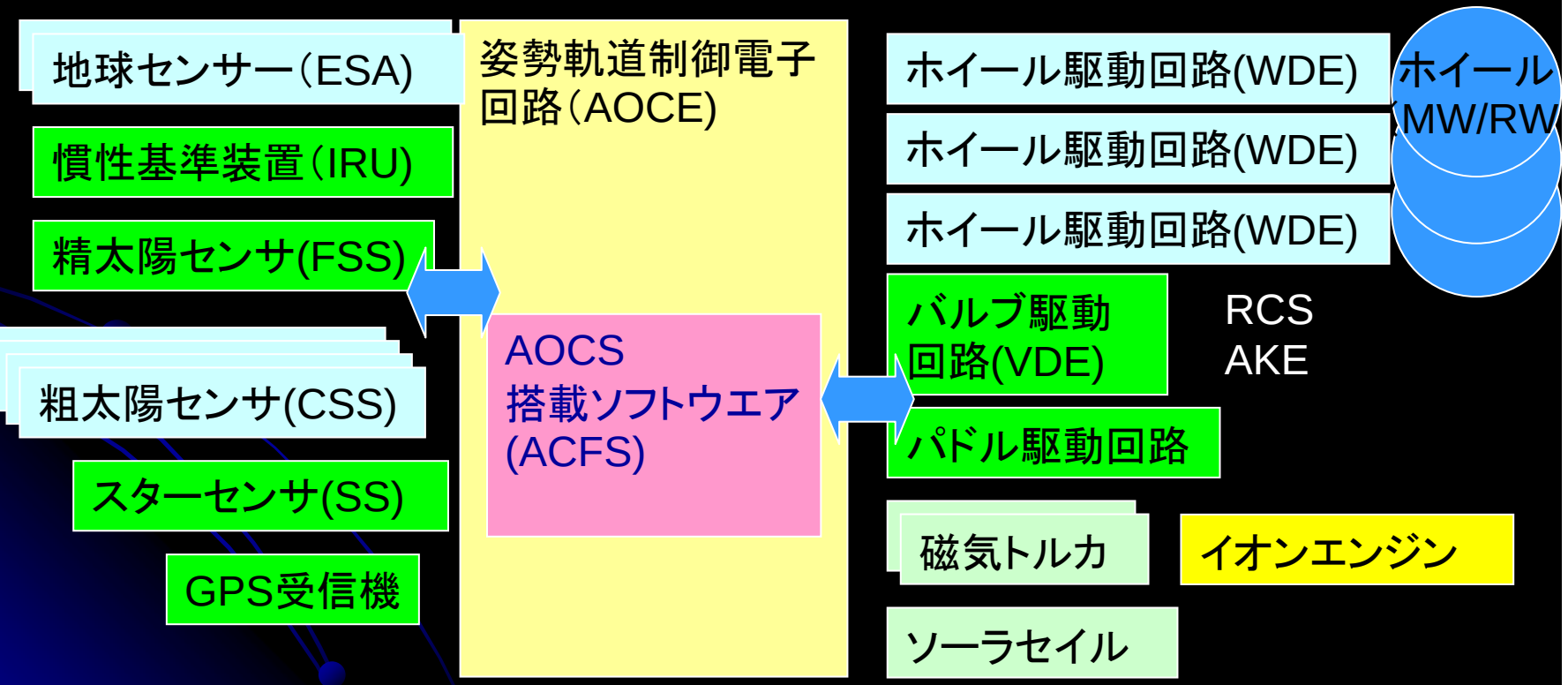
- 人工衛星
  - 姿勢制御系、データ処理系、ミッション系、スタートラッカー／GPSRなどセンサー系など組込み系が中心
  - Fail Safe → Fail Operative
  - FDIR: Failure Detection Isolation and Recovery
  - 最近は書換可能のものもある
  - アセンブラ、C言語
  - 再利用／シリーズ化
  - 放射線の影響

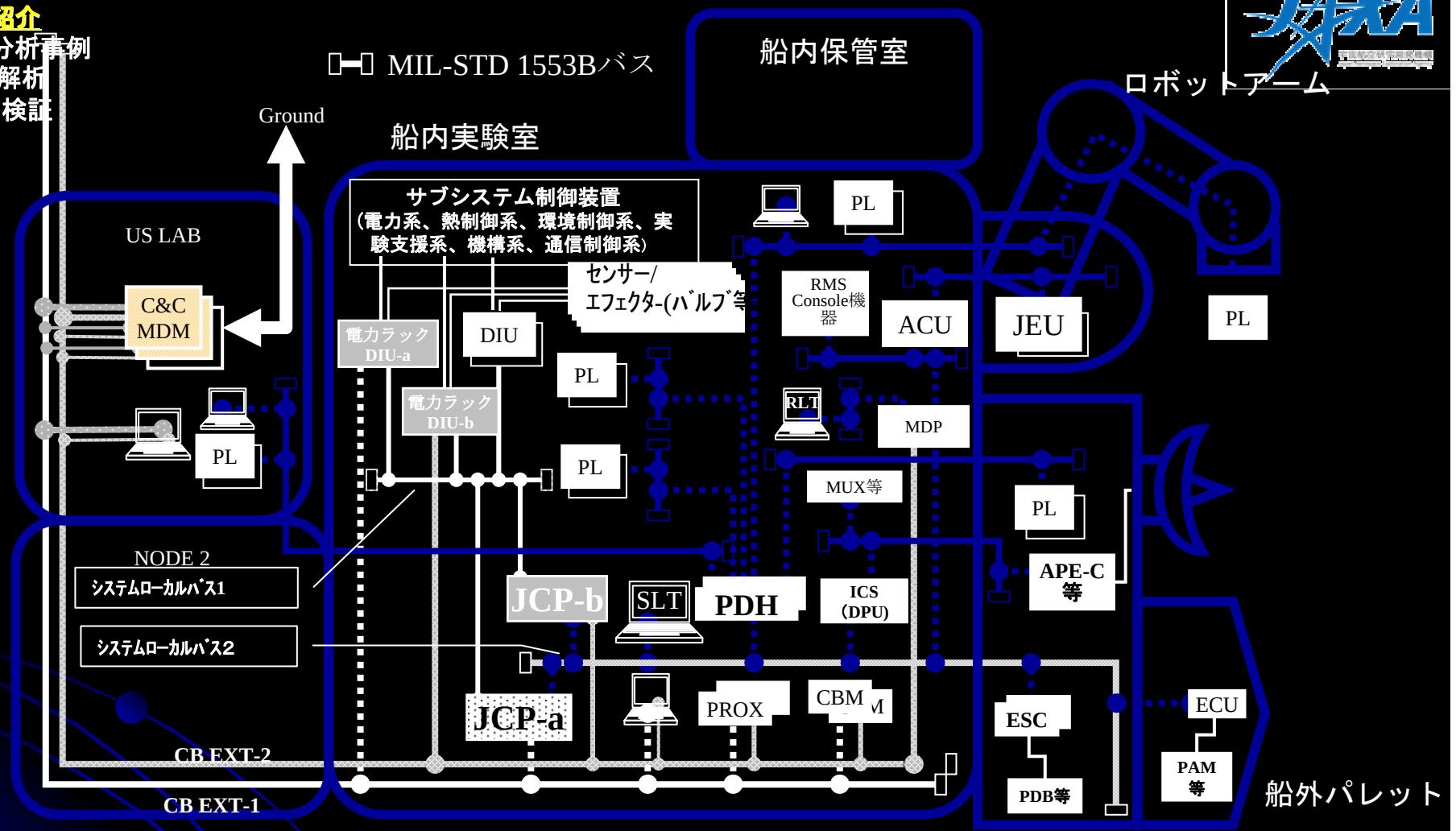
- 宇宙ステーション
  - 管制システム、実験装置など膨大な数のソフトウエア
  - 高い安全要求（2故障でも安全に！）
  - コマンド、データ、処理の独立性
  - 複雑で膨大なFDIR
  - システム構成にクルーが入る
  - Ada、C言語

- ロケット
  - 航法誘導制御系など
  - 高信頼性／多数決
  - アセンブラ、C言語
  - ハードリアルタイム
- 地上管制システム
  - 管制システムなど
  - 安全性

衛星アビオニクスの機能と構成

- 姿勢/軌道制御、通信及びデータ処理(システム、ミッション)、FDIR
- 姿勢軌道制御系(AOCS)





JCP:管制制御装置
SLT:きぼうラップトップ端末
PDH:実験データ処理装置
DIU:データインタフェースユニット
MDP:RMS管理計算機
ACU:アーム制御計算機
JEU:関節エレクトロニクス

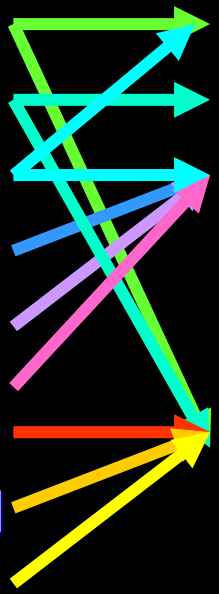
ICS:衛星間通信システム
DPU:データ処理装置
APE-C:アンテナ駆動装置
ESC:船外実験プラットフォーム制御装置
ECU:船外パレット制御装置
PAM:船外パレット装置交換機構

JEM管制系系統図

特徴と対策

信頼性・安全性が何よりも優先

- 高信頼性
- 安全性（≠信頼性）
- 開発チーム・企業数が多い
- ハードウエア主体の開発
- 繰り返し開発と単品開発
- 長開発期間と要求仕様変更
- 特殊環境（放射線、ノイズ）
- 自律運用とオペレータ協調運用
- 軌道上書換え



- 検証プロセス強化
- 安全要求と安全審査プロセス
- 開発プロセス・組織体制
- 高信頼性・安全性設計

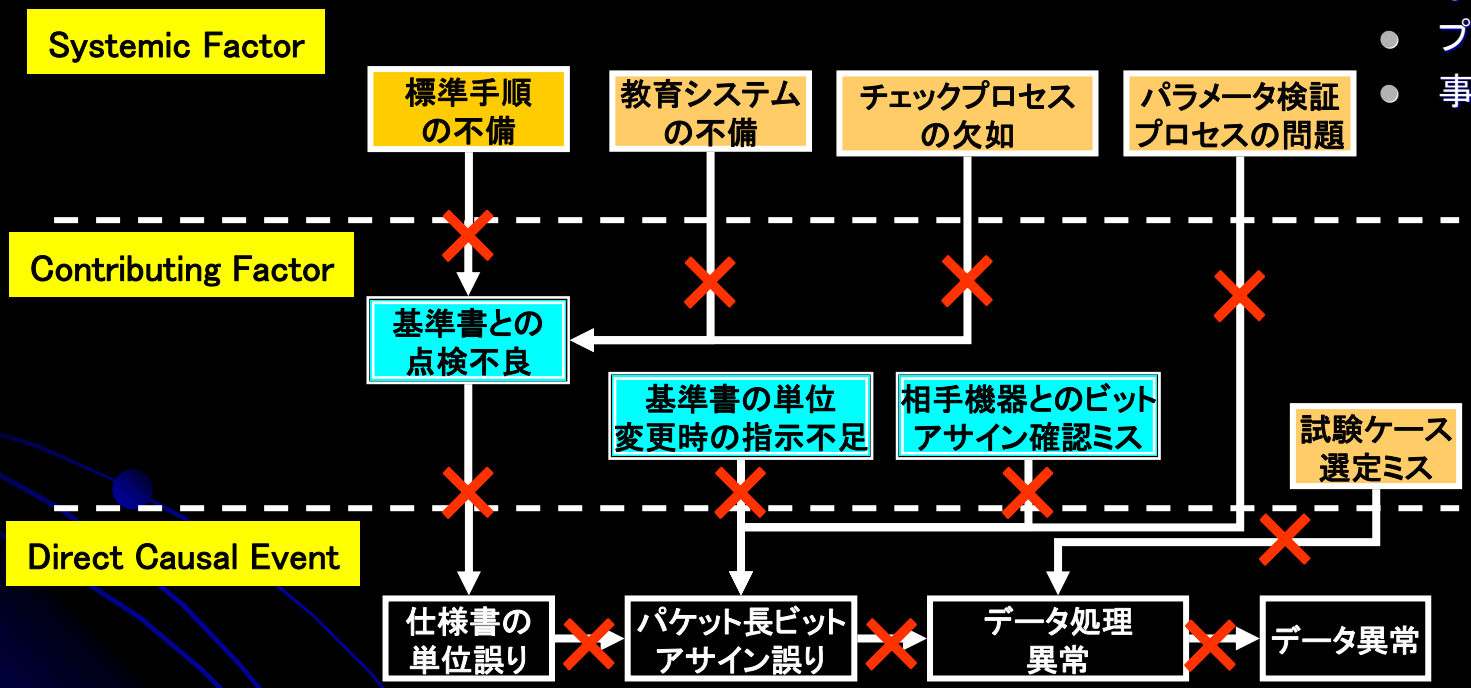
開発・検証プロセス上の課題を発見するために....

事故モデル分析

- 事故モデル分析*1を用い、不具合に対するプロセス課題を分析

*1: MIT Nancy Leveson 教授 考案

- 分析技術利用範囲
  - 事故分析
  - プロセス要因分析
  - 事前安全評価



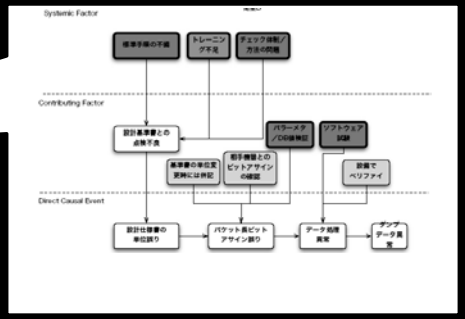
分析前不具合原因
分析前不具合対策範囲
分析で新たに識別されたプロセス課題

× 何処で連鎖を切っても不具合は防御できる
不具合分析を通じて改善箇所の優先準備を決める

注意点：
- 技術担当者が参加
- 分析技術習得者が参加
- 分析のための分析にならないように

SEPG Japan 2004

ポストモーテムによる2次分析



１次プロセス要因分析

事故モデル分析

２次プロセス要因分析

ポストモーテムプロセス

原因抽出

原因項目分類

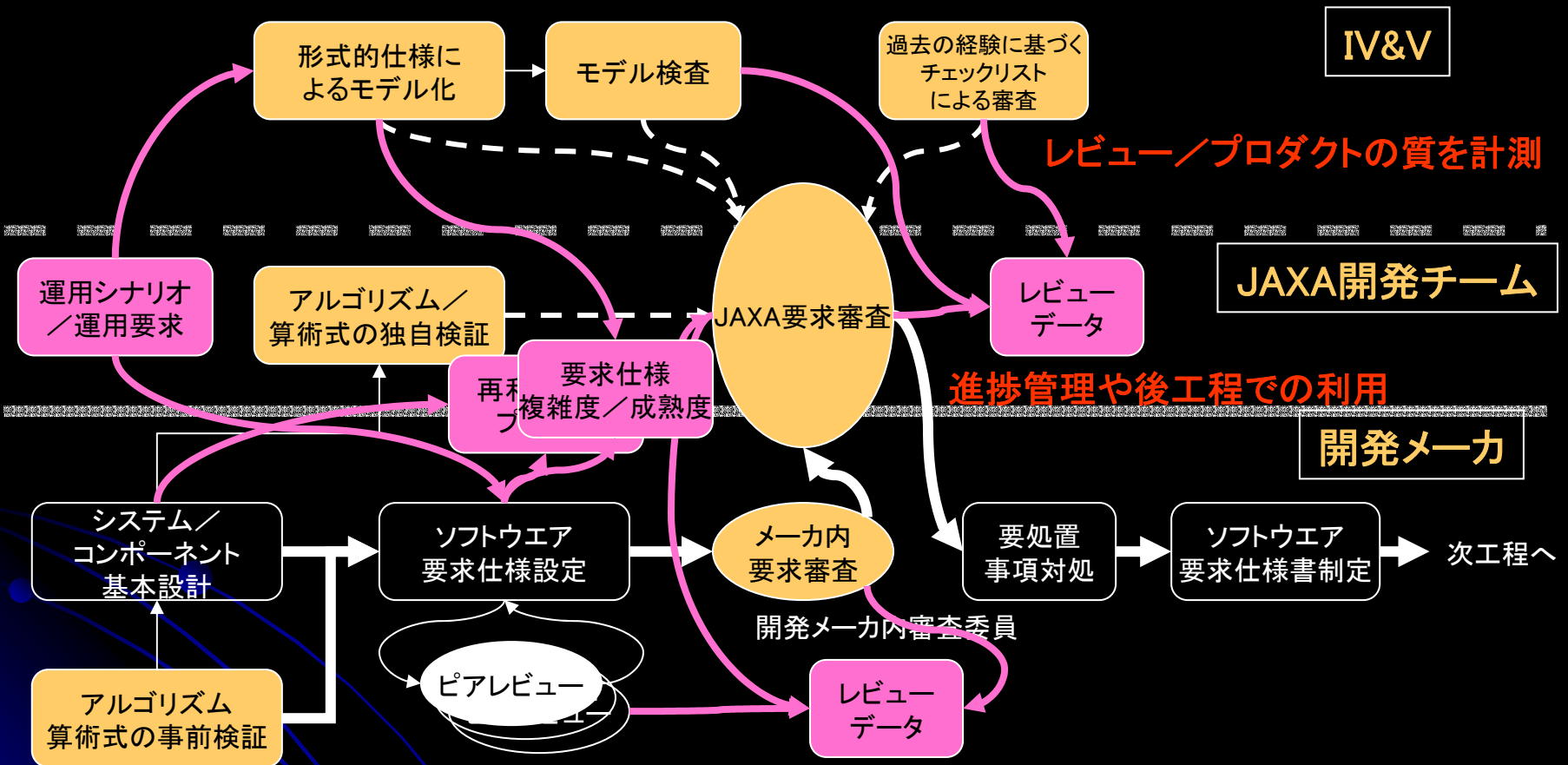
原因因果関係分析

根本原因・対策検討

プロセス上の課題 （追加シート）

- JAXA、メーカの作業プロセスの連携が不十分、責任が不明確（ケースバイケースというトリック）
  - ⇒ 統合プロセス（開発メーカ・JAXAの関係を含めた）の定義
- SQA要求に対する開発チーム、開発メーカの不満
- 凝り固まったプロセス・形骸化
  - ⇒ IV&Vの導入
  - ⇒ 定量的な分析ポイントの増強

開発プロセスの改善事例



通常、EM(Engineering Model)及びPFM(Proto Flight Model)の2段階のステップで開発が実施される、または、開発要素／進捗により段階的(Iterative)なステップで開発が実施されるため、要求仕様の差分(Δ)を再度同じ工程で審査することになる。

設計・解析上の工夫として….

安全性・信頼性設計

- 部品からシステムレベルまで、設計、製造プロセス、試験・検証を通じて品質を積み上げる。(H/W、S/W)
- ハードウェアの冗長系を構成して故障時に切り替えることにより信頼性を確保する。
- ソフトウェアによるFDIRにより信頼性を確保する。
  - ロケット：基本的にシングル（冗長系なし）、多数決
  - 人工衛星：1フェイルトレラント
  - 宇宙ステーション（有人）：2フェイルトレラント
- 宇宙飛行士、地上運用者による不具合発見と対処

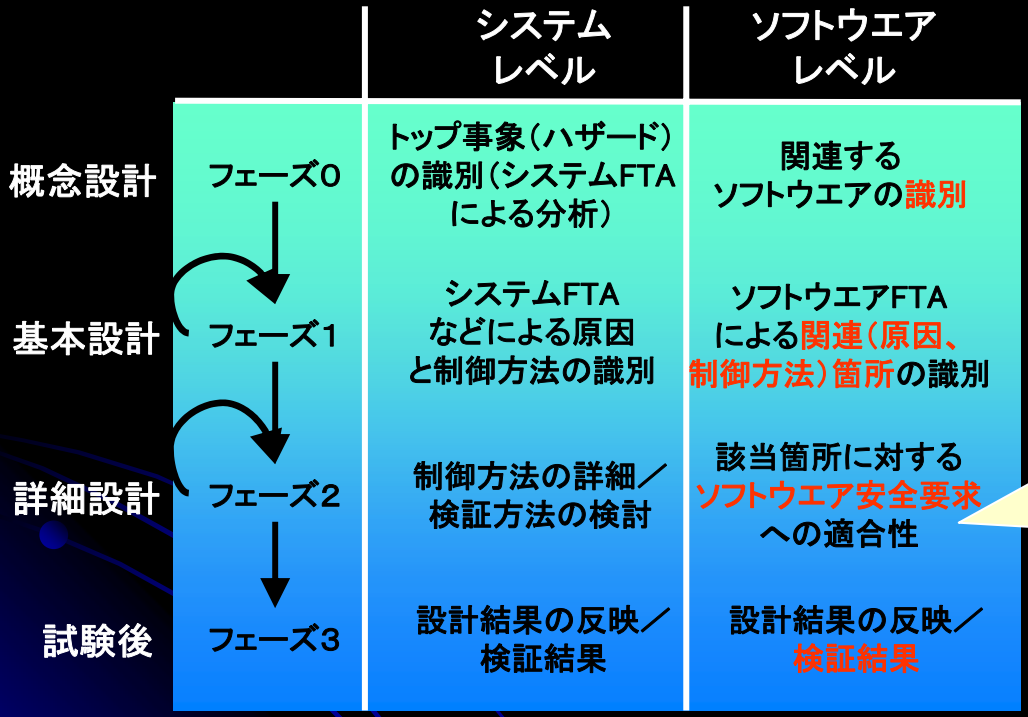
- 安全性・高信頼性のための設計
  - 安全設計
    - 2fail safe
    - Safing
    - 縮退モード
  - 信頼性設計
    - 故障許容設計
      - 冗長系
      - FDIR
      - 多様性
      - 軌道上書き換え

安全性・信頼性設計

- 安全性・高信頼性のための解析
  - ソフトウエアハザード解析／ソフトウエアFTA（トップダウン解析）
  - ソフトウエアFMEA（ボトムアップ解析）
  - ソフトウエアフォールト伝搬解析（水平伝播解析）
    (フォールトインジェクション)

安全審査プロセスとソフトウエア安全要求

- 安全審査 プロセス（開発フェーズに沿った重要プロセス）



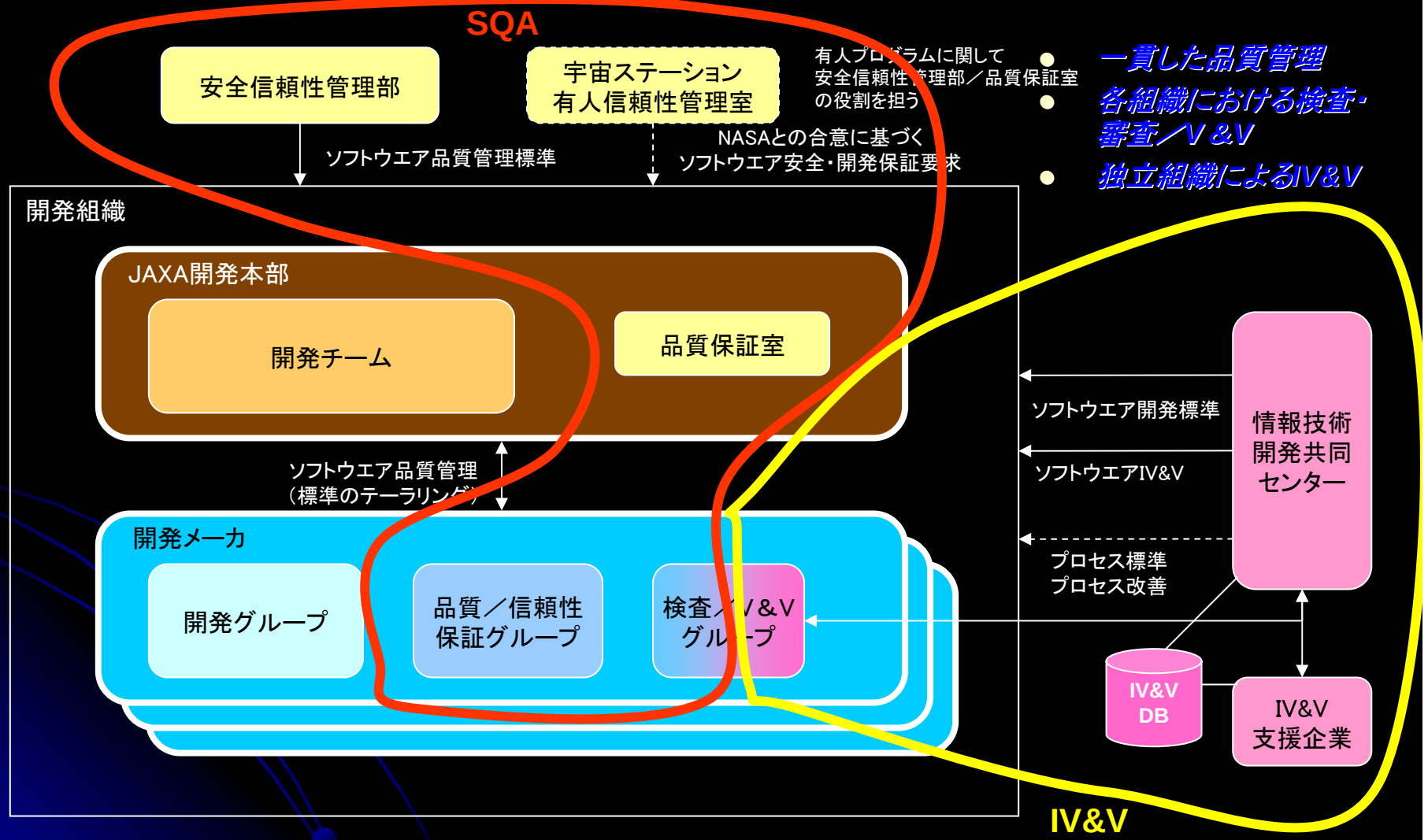
ソフトウエア安全要求
（宇宙ステーションの例）

- 国際間合意基準

1. 全般要求
2. Must Not Work機能要求
3. Must Work機能要求

第三者による綿密なレビュー
・開発メーカ内
・JAXA内（設計担当、安全担当、QA、IV&V）
・NASA

宇宙用高信頼性ソフトウエアの開発体制

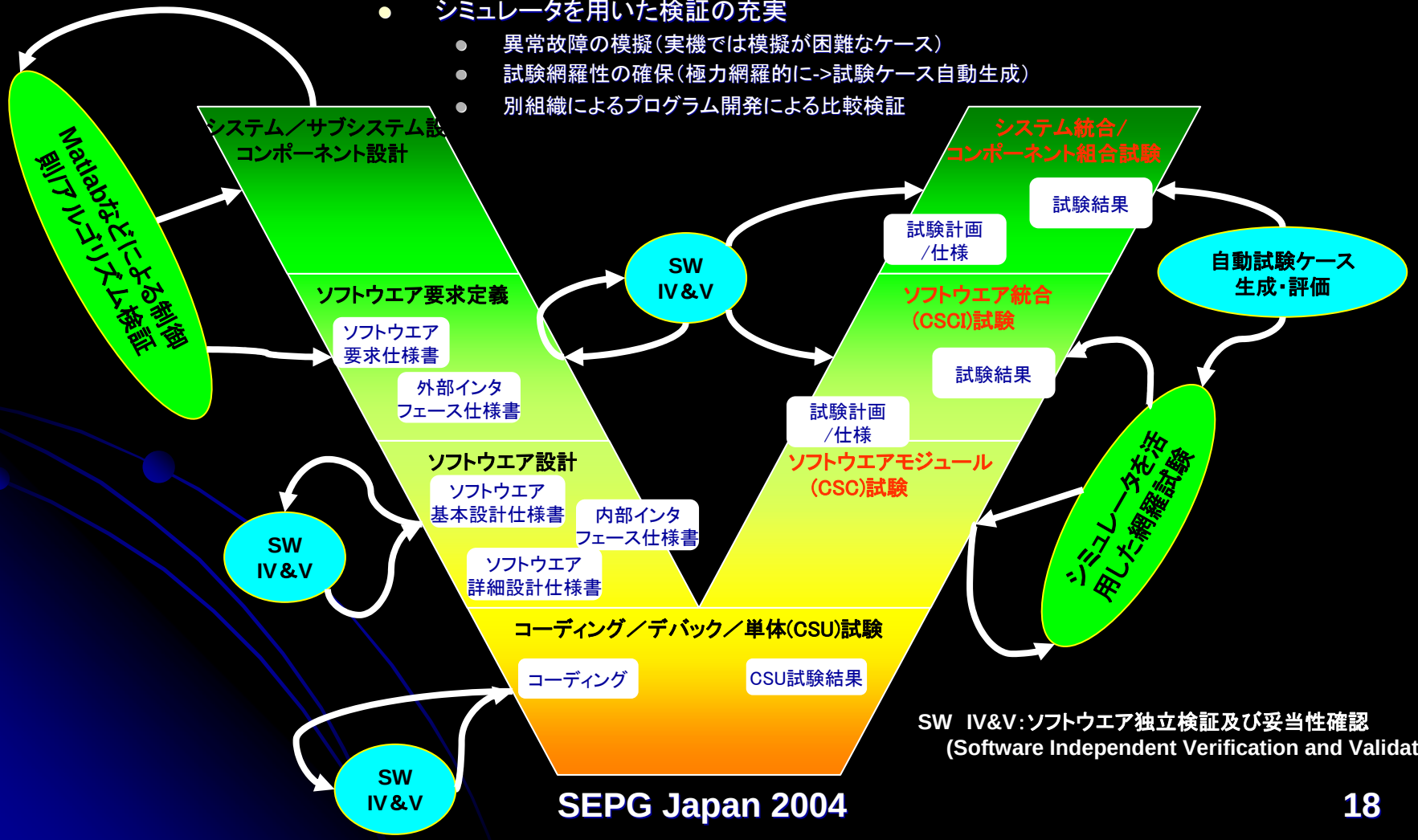


- 一貫した品質管理
- 各組織における検査・審査／V&V
- 独立組織によるIV&V

独立組織による複数のチェック体制

高信頼化のための検証プロセス

- ソフトウエア開発初期段階におけるモデル評価
  - Matlabなどを用いた制御アルゴリズムの事前評価
  - 形式的仕様記述モデルを用いた分析
- シミュレータを用いた検証の充実
  - 異常故障の模擬（実機では模擬が困難なケース）
  - 試験網羅性の確保（極力網羅的に->試験ケース自動生成）
  - 別組織によるプログラム開発による比較検証



SW IV&V：ソフトウエア独立検証及び妥当性確認
(Software Independent Verification and Validation)

ソフトウエアIV&Vとは…

SW IV&Vプロセス

- シャドウプロセスとしての役割

【目的】

・新規技術・プロセスの実証

・不具合・問題点の発見

【実施内容選定】

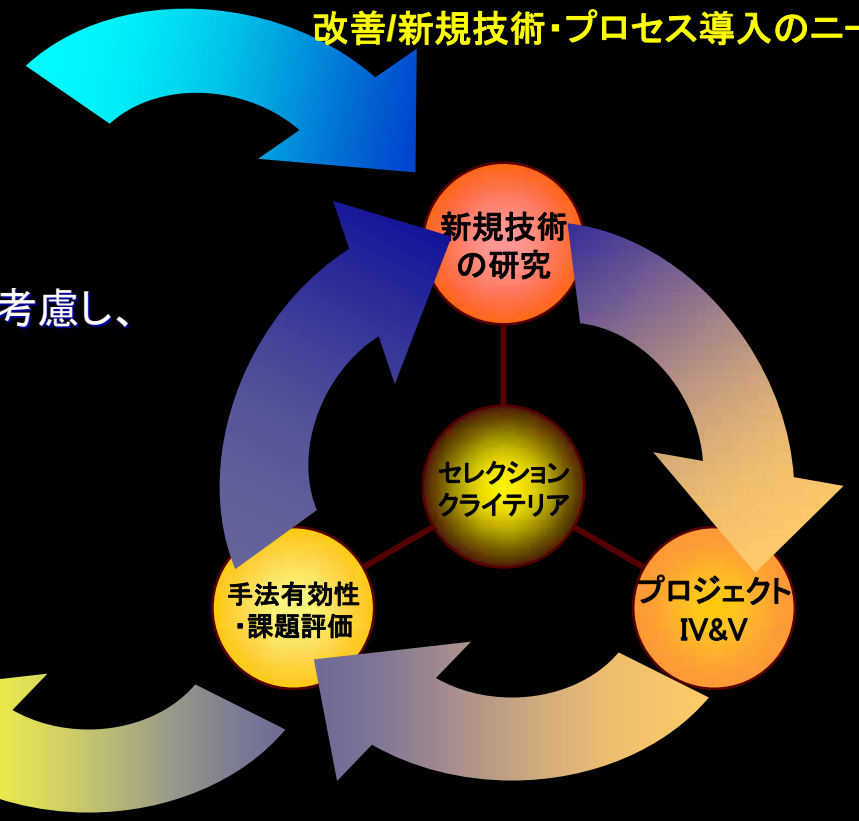
各プロジェクトのIV&V実施にあたり、以下を考慮し、実施内容を選定・決定する

・不具合定量分析による重点箇所
・プロジェクト内作業者インタビュー
・ソフトウエアの特徴・特性
・データアクセス性
・許容工数とスケジュールによる優先順位付
・作業開始時期

【手法の有効性評価】

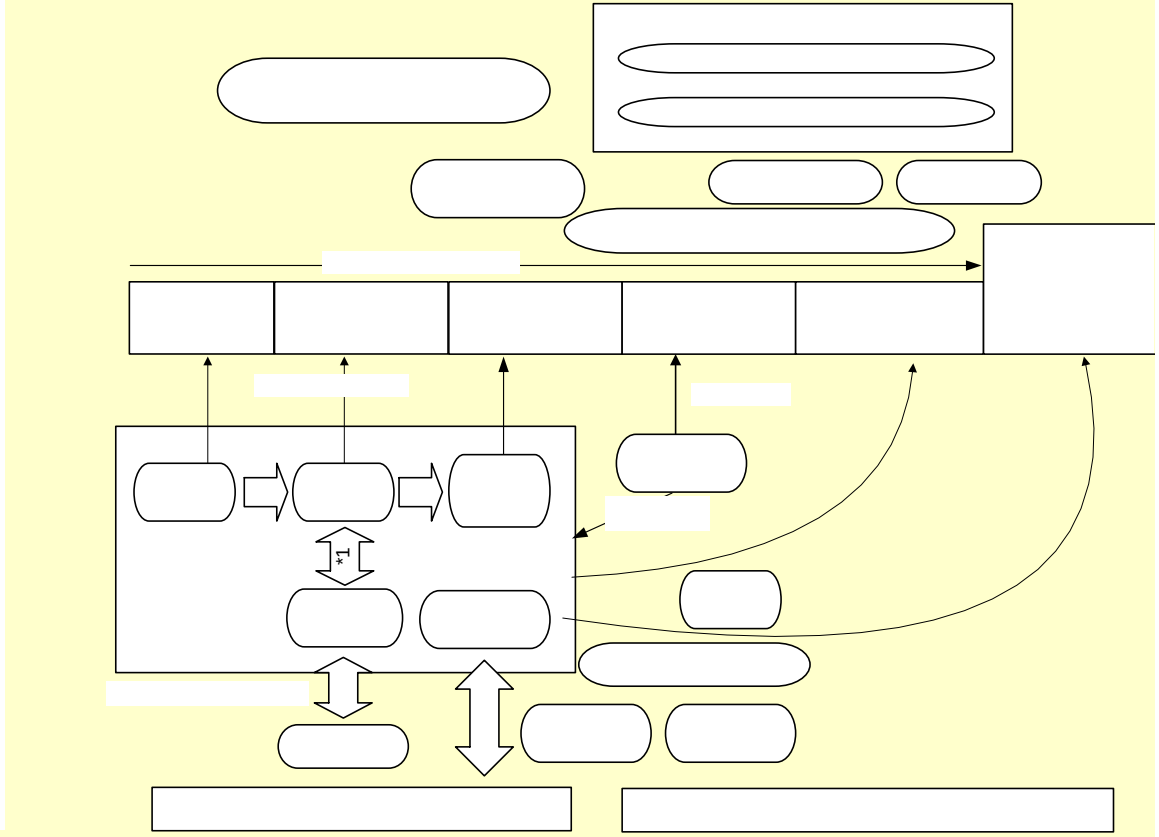
有効性を以下の基準で評価し、次プロ
映する。

・問題点発見総件数
・重要問題点摘出比率
・作業工数
・手法利用時の課題
・成果に対するカスタマーの評価



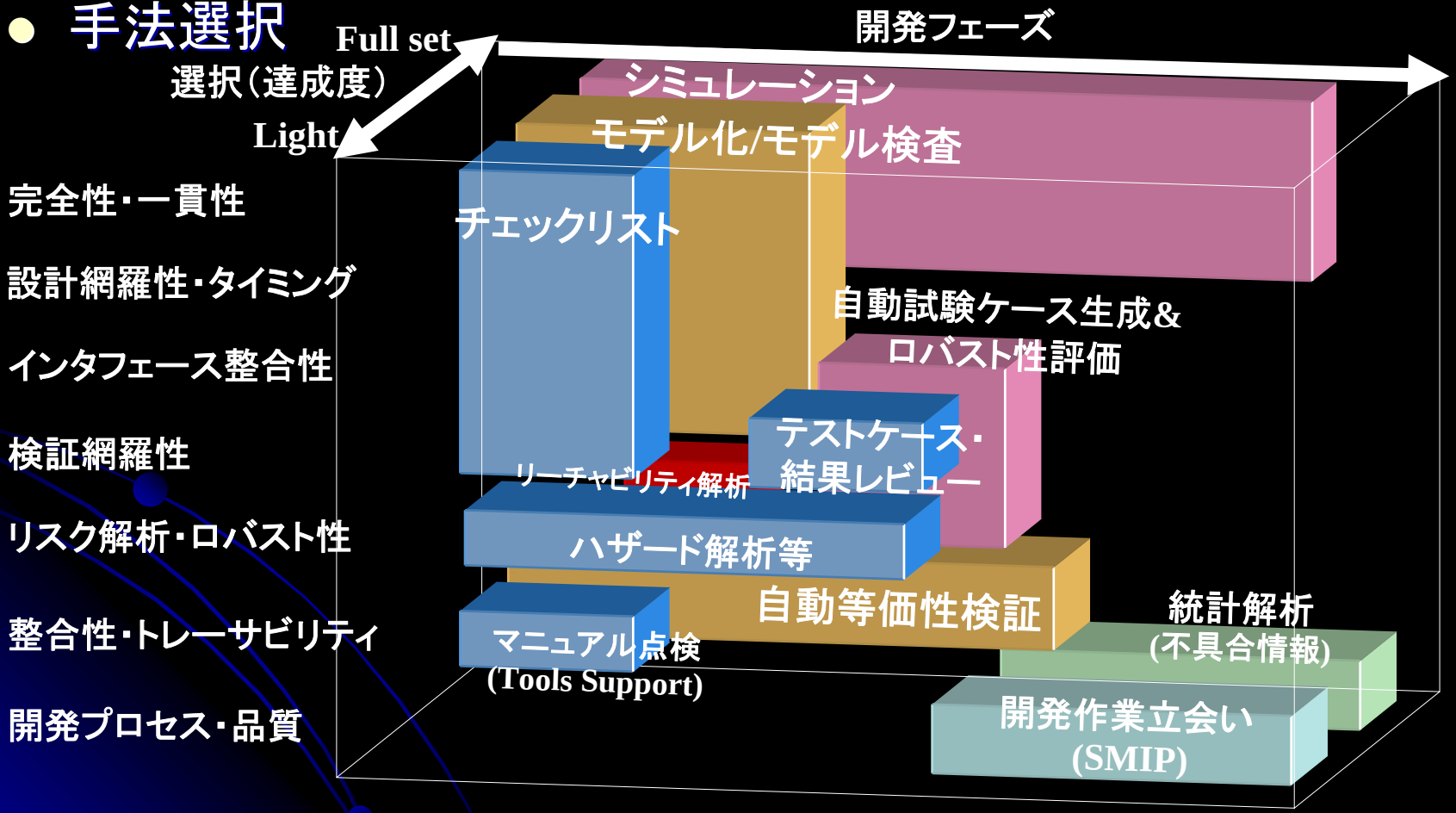
開発プロセスの沿った ソフトウエア(SW)IV&Vプロセス

解析評価手法名 ＼ 開発フェーズ	要求分析	基本設計	詳細設計	コーディング	結合試験	総合試験・実地検証	運用
1.モード遷移解析	○	○	○				○
2.完全性解析	○	○	○				○
3.ノミナルシミュレーション					○	○	○
4.FDIR（1故障時）網羅性解析	○	○	○				○
5.ソフトウェアデビエーション解析		○	○	○			○
6.単一故障シミュレーション				○	○	○	○
7.FDIR 階層分析	○	○	○				○
8.複数 FDIR 並列実行解析	○	○	○	○	○	○	○
9.安全モード遷移解析	○	○	○	○	○	○	○
10.二重故障シミュレーション					○	○	○
11.HAZOP 解析					○	○	○
12.一貫性解析	○	○	○				○
13.共有変数解析				○			○
14.コードクローン解析				○			○
15.階層モデルによる事故事例解析	△	△	△	△	△	△	△
16.自然言語仕様によるモデルの自動生成	○	○	○	○	○	○	○
17.トレーサビリティ解析	○	○	○	○			○
18.リーチャビリティ解析	○	○	○				○
19.運用制約識別	○	○	○	○	○	○	○
20.インターフェイス解析	○	○	○				○
21.タイミング解析		○	○				○
22.二故障安全網羅テストケース解析			△	○	○	○	○
23.姿勢制御系搭載ソフトウェアチェックリスト	○	○	○	○			○



手法の選択(例)

- 手法選択


Full set
選択(達成度)
Light
開発フェーズ
完全性・一貫性
設計網羅性・タイミング
インタフェース整合性
検証網羅性
リスク解析・ロバスト性
整合性・トレーサビリティ
開発プロセス・品質
シミュレーション
モデル化/モデル検査
チェックリスト
自動試験ケース生成&
ロバスト性評価
テストケース・
結果レビュー
リーチャビリティ解析
ハザード解析等
自動等価性検証
マニュアル点検
(Tools Support)
統計解析
(不具合情報)
開発作業立会い
(SMIP)

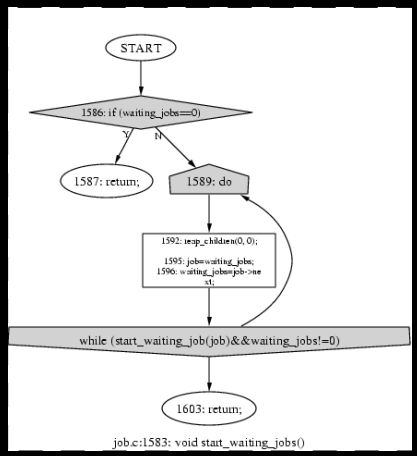
ソフトウエアIV&V手法の事例…

IV&V手法事例
（上流工程・モデルベース）

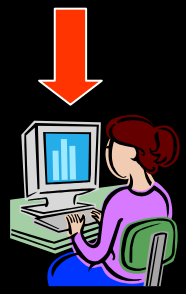
仕様書

AA Mode

A) The SYSTEM XYZ shall (REQ-01001) transfer mode from every mode without Initialization Mode or B Mode to AA Mode by command.



文書(フロー図)



自然言語モデル
(SpecTRM Model)

Modeling

SpecTRMモデル

AA Mode (REQ-01001)
'XYZ is Stanby_Mode' and 'AA_Mode_Command is ON'
Or 'XYZ is AA_Mode' and 'AA_Mode_Command is ON'
Or 'XYZ is AB_Mode' and 'AA_Mode_Command is ON'
=> XYZ is AA_Mode

- 一貫性解析
- 完全性解析
- トレーサビリティー解析
- 等価性解析
- リーチャビリティ解析
- IF解析 等

フローチャートをXML Model ファイルからC言語のプログラムの形に自動変換

Cコンパイラでコンパイルし状態モデルを生成

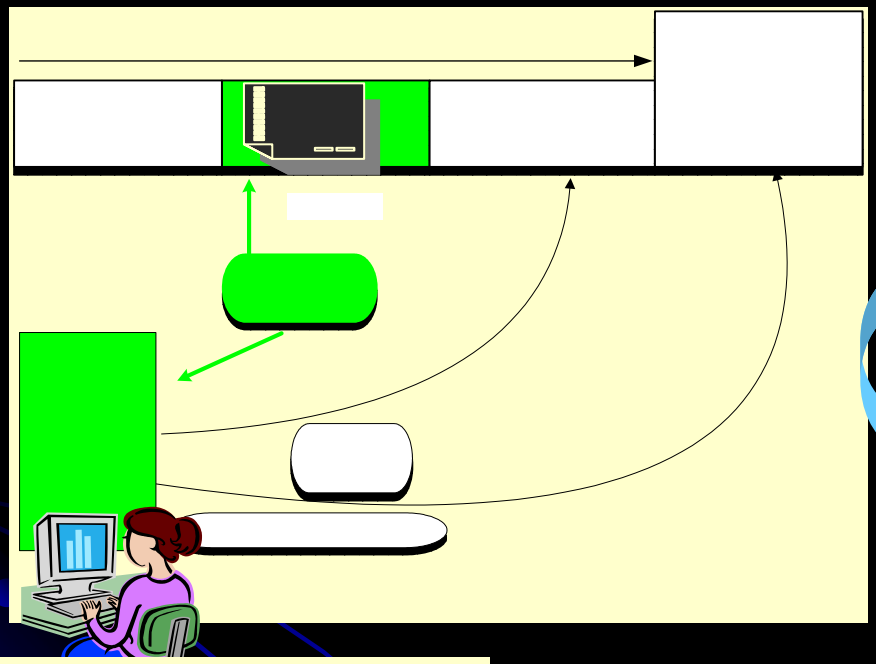
シミュレーション

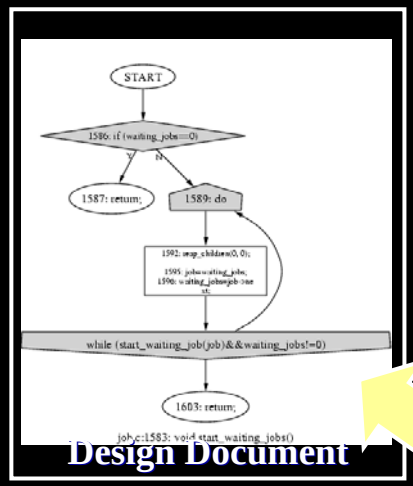
- 実行されないフローをチェック
- 変数の初期値設定チェック
- 条件分岐の一貫性
- ゼロ除算の可能性 等

トレーサビリティー/等

IV&V手法事例(下流工程 ツールチェック/コード一詳細設計書の比較)

★詳細設計書、ソースコードを用いたアプローチ





Design Document

等価性検証

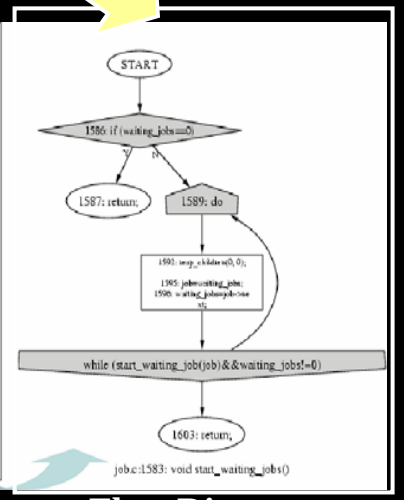
自動検査アルゴリズム

```
void start_waiting_jobs ()
{
 struct child *job;
 if (waiting_jobs == 0)
  return;
 do
  {
   /* Check for recently deceased descendants.  */
   reap_children (0, 0);

   /* Take a job off the waiting list.  */
   job = waiting_jobs;
   waiting_jobs = job->next;

   /* Try to start that job.  We break out of the loop
as soon
     as start_waiting_job puts one back on the
waiting list.  */
  }
 while (start_waiting_job (job) && waiting_jobs != 0);
 return;
}
```

Source code



Flow Diagram

コードチェックツールの活用により、
・上流工程の質を測定することが可能となった。
・不具合の予測に利用できた。
・試験計画に利用できた。
⇒IV&Vから主開発プロセスへ

活用のポイント
・評価対象メトリクスデータの絞込み
・適用ルールの絞込み
・結果評価プロセスの設定

Tool Check
・静的解析
・動的解析

・コードへのトレーサビリティ検証
・設計網羅性検証
には、有効であった

IV&V手法研究
(下流工程・試験ケース分析)

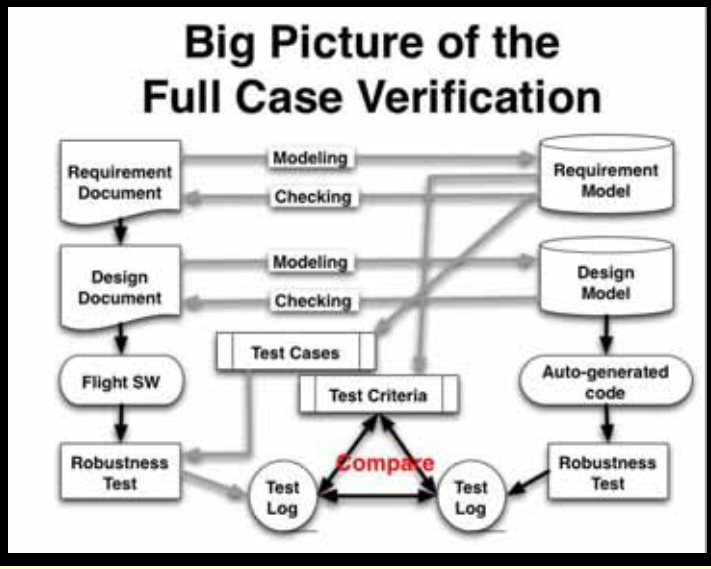
- 限られたリソースの中で、ソフトウェアテストにおいて全てのケースを流すことは不可能
- ソフトウェアの観点からシステム的に問題がないことを証明することが必要
- ➲ 同一パターンにまとめることにより、テストケースを削減をすることが可能
  - 例） 2故障許容のシステムに対して：
    - 2故障が独立な場合は、1故障の試験のみを行えばよい
    - あるフェーズで起こりえないケースに関しては、テストをする必要はない

テストケース除去のイメージ

: 削除されるテストケース

ケース削減から
自動ロバスト試験へ

形式的仕様モデルを用いた試験ケース
自動生成と自動サクセス比較



・人が介在しないことで、網羅的なロバスト試験が可能となる
・適切な試験ケース生成のためのアルゴリズムが鍵

まとめ

- 事故モデル分析などを活用しプロセス上の課題を効果的に発見できた
- 信頼性・安全性向上のためには、通常の開発プロセスに加え、特別な検証プロセスが必要
- 特にモデルによる解析及びソフトウエアIV&Vは、適切な計画・実施により、十分な成果を達成することができる

今後 （追加シート）

- 製品・プロセス品質の計測を強化
- 上流工程の改善
  - 主開発工程にモデルベース開発を導入
- 開発メーカ・JAXAを網羅した統合プロセスの定義