高信頼性。ソフトウエアテスト自動化ツール

航空機業界で採用される
コーディング標準、ユニットテスト、モデル検証手法のご紹介

LDRA Software Technology

TVEC

We Cover All Boundaries

FUJI SETSUBI

テストに関するデータ

- 企業はテストに75%もの時間を費やしている
  - テスト担当者によるテスト時間がその50%
  - 開発者の50%の時間がユニットテストに（全体の25%）
- システムが複雑になるにつれ、欠陥の検出は一層困難に
- テストの開発に相当な費用がかさんでいる
  - 良いテストをデザインするツールが少ない
  - テストそのものの開発や、そのテストにも時間が費やされる

（テスト期間の半分はテストスクリプトのデバッグ）

LDRA Ltd

Liverpool Data Research Associates

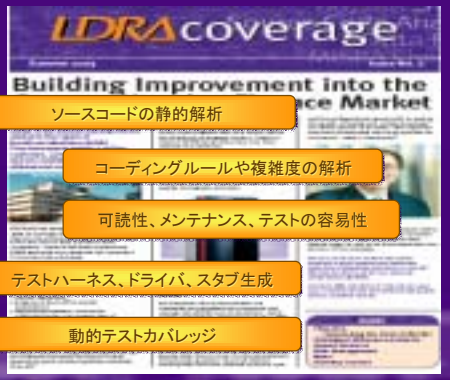
Founded 1975

Provider of test tools & solutions

Metrics Pioneer

Consultancy, support & training

開発期間の25%、テストの75%を削減



ソースコードの静的解析

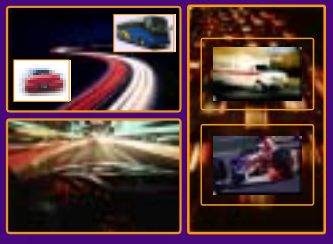
コーディングルールや複雑度の解析

可読性、メンテナンス、テストの容易性

テストハーネス、ドライバ、スタブ生成

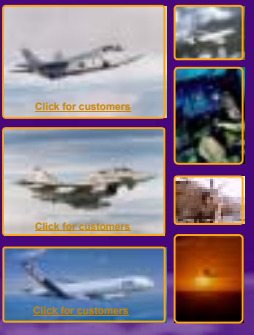
動的テストカバレッジ

LDRA on the 自動車



LDRA - 'Helping to get through the Test'

LDRA in the 航空



LDRA on 鉄道







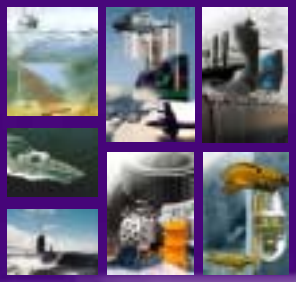
LDRA - 'Testing along the right lines'

LDRA in 通信機器




LDRA - 'Sound Testing Practice'

LDRA at 船舶



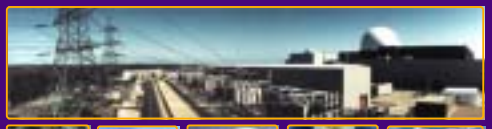
LDRA 'Testing the Water'

LDRA in the 金融システム

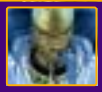



LDRA – 'Testing to Bank on'

LDRA in 電力システム








LDRA - 'Power Tools'

LDRA in 宇宙機器






LDRA – 'Expanding the Frontiers of Testing'

LDRA in 医療機器

PHILIPS
GAMBRO
novo nordisk
GRASEBY
ESAOTE
HOSPAL
GE Medical Systems

'Software health check'

Back

LDRA Tool Suite Languages – LDRA Testbed

Languages:

C
C++
C#
Ada 83 / Ada 95
Java
Visual Basic 6
Visual Basic.net
Cobol
Coral66
Fortran
Pascal
PL/Mx86
PL/1
Algol

Assemblers:

intel
MOTOROLA
Texas Instruments
PowerPC

LDRA Tool Suite Host Platforms

- Windows 9x/NT/2000/XP
- HP OpenVMS (Alpha & IA-64 Itanium)
- VAX VMS
- MVS
- z/OS

Unix

Linux
Sun Solaris
Digital
AIX
SGI IRIX
HP-UX

Unisys

A/ClearPath/MCP
os 2200

LDRA Tool Suite for Embedded System Testing

Development Tools

Ashling Vitra, Pathfinder & asIDE
AONIX ObjectAda
ARM ARMulator / Jumpstart
Cosmic ZAP
Green Hills MULTI & INTEGRITY
Hitex ICE
IAR
KEIL
Lauterbach ICE
Metrowerks CodeWarrior
Microtec XRAY
Nohau Emulators
TASKING
TI CodeComposer
WIND RIVER Tornado & Diab

Processors

ARM
ARM7/ARM9
Infineon
TriCore
Intel
8051, 80C196 & others
MIPS
SmartMIPS, HiPerSmart, 4K(x)
Motorola
MC68K, MPC5xx & 6xx
TI
TMS320Cxx
PowerPC
603, 603E, 604 & others
& many more…

Customer Profile

- Users include major national and multinational companies and agencies of British, American and other governments:

LDRAによるテストの自動化

開発期間の　81%　をサポート

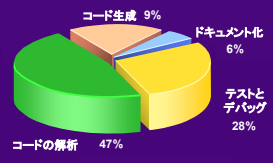
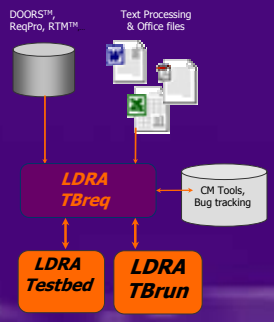


Figure 1: How Developers Spend Their Time

Source: Software Quality: Producing Practical, Consistent Software
Mordecai Ben Menachem & Garry S. Marliss. Thomson Computer Press 1998

LDRA　開発ライフサイクルにおける自動化促進

LDRA Test Manager adds TBreq



LDRA Platform Test Plan Support

Requirements Verifica n
- Black Box testing/Functi ting/High-Level Reqs
- Conforms to DO-178B B & C)
- Required for CMMI ve
- Applicable to any nts regime

Path Testing
- All feasible mation
- White Bo
- Confor
- Broa
- Cri

Your *ZDSD* Test Plan

TESTED

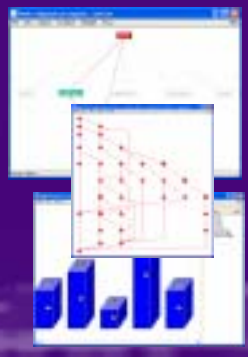
- Safety Critical Software Plan
- Mission Critical Software Plan
- Name your plan!!!

多くのスタンダード、ガイドラインをサポート

- DO-178B (Levels A, B & C)
- DEF STAN 00-55 (Static & Dynamic)
- MISRA C / MISRA-C:2004
- DERA C
- BS 7925
- IEC 61508 (SIL 4 – 1)
- CENELEC 50128
- BS EN ISO 9000 / 9001:2000
- NUREG 6501

静的解析　:コードレビュー機能

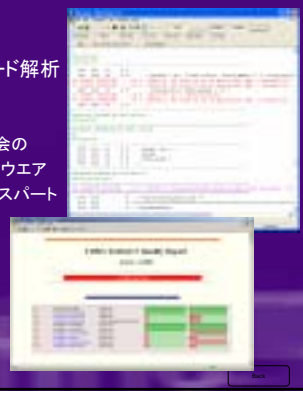
- プログラミングスタンダード
- コードレビューのレポート
- コード構成のグラフィカル表示



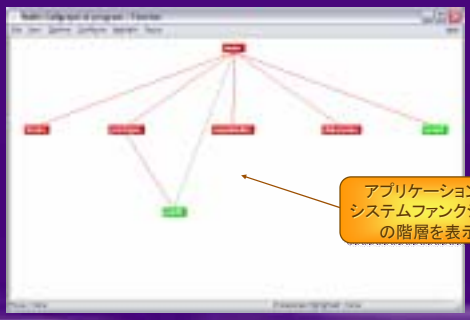
MISRA C / MISRA-C:2004 を完全に対応

MISRA C /
MISRA-C:2004 スタンダード解析

– LDRAはMISRA C 委員会のキーメンバーであり、ソフトウエアテストとそのツールのエキスパートとして広く認知されている

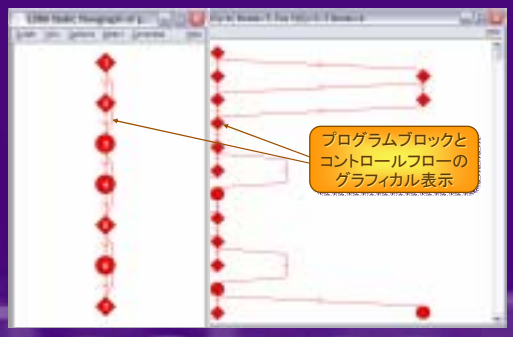


コールグラフによるコード構成のグラフィカル表示

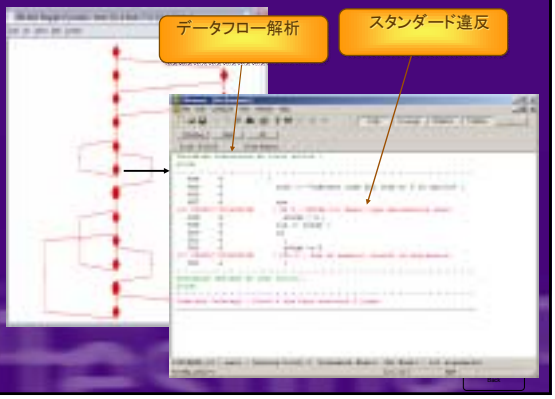


赤は、アプリケーション
緑は、アプリにコールされるシステムファンクション

フローグラフによるコントロールフローのグラフィカル表示

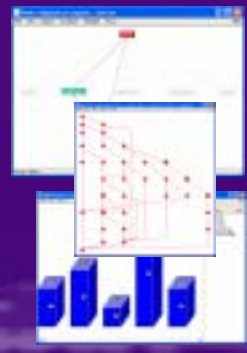


各デシジョンノードごとのソースコード解析表示



クオリティーレビュー機能

- 複雑度解析
- リファクタリングサポート
- LCSAJs　パス解析
- 品質レポート



複雑度解析　:グラフ表示

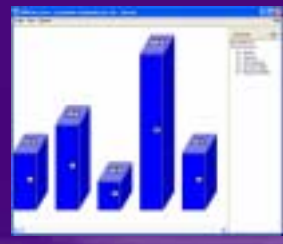
- より良く構造化されて、複雑すぎないか？
- メンテナンスは容易であるか？
- 問題が起こり得そうな場所は？

直感的なグラフ表示で解析

複雑度解析　:各種メトリクスの統計

複雑度解析は、プロシジャーごとに解析され、その構造をレポートする

各種複雑度のメトリクスは、プロシジャー、ファイル、システム全体等に対して評価、解析される

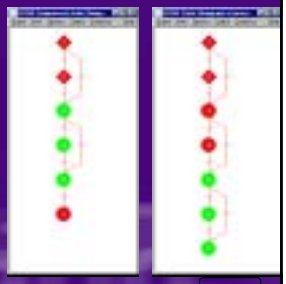


SPV 本質的な複雑さを明示　リファクタリングを促進

Structured Programming Verification (SPV) により、複雑度とノットメトリクスは適性に評価される
本質的に複雑である箇所の特定

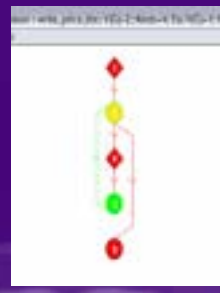
グラフィカルに解析

本質的に複雑である箇所の特定をし、不適切に構造化された箇所のリファクタリングを促進

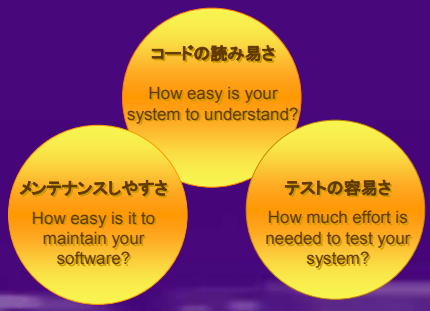


LCSAJs　パス解析

- LCSAJs 各コード行が持っている実行パスを解析
- コードのメンテナンス性の重要なメトリクス
- テスト不可能なパスも特定
- 最も包括的なパスカバレッジ（テストカバレッジとの融合）
- Mandated for European safety critical military avionics projects



品質の概要表示　:各メトリクス結果を統合して

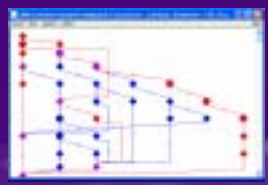


品質の概要表示（各メトリクス結果を統合）

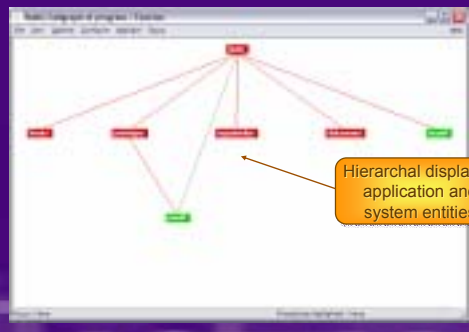
A representation of bounds tested quality metrics. Bounds can be user defined or industry standard.

デザインレビュー

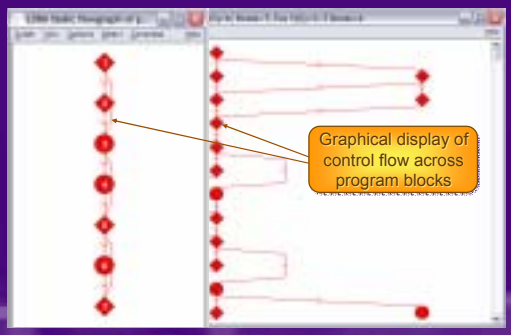
- Requirements Traceability
- Interface Analysis
- Coupling Analysis
- Data Anomalies



Code Review – Code Visualisation – Static Callgraph



Code Review – Code Visualisation – Static Flowgraph



Design Review – Requirements Traceability

A vitally important aspect of software development is that “the design be implemented as required”

- Requirements traceability must be verified
- TBreq offers LDRA / DOORS integration

TBReq can analyse interfaces and verify them against specifications

TBreq can also map requirements to file, procedure and statement block levels

Design Review – Coupling Analysis

- **Nuclear Spider**
  - Procedures that either call or are called from the node
- **Extended Spider**
  - Procedures that are either the nodes ancestors or descendants
- **Data Coupling**
  - Parameter and variable usage

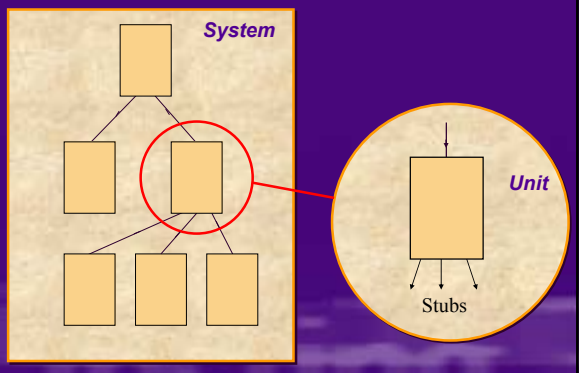
Design Review – Data Anomalies I

Data flow anomalies are sequences of actions related to a program variable which are suspected to be erroneous.

Messages produced by the Data Flow Analysis report detect different types of data flow anomalies.

Data Flow Analysis is performed across procedure boundaries in the source file or system-wide.

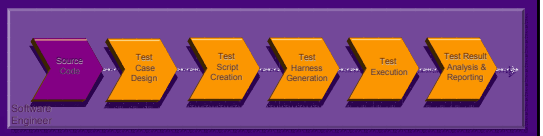
ユニットテスト



Unit Test

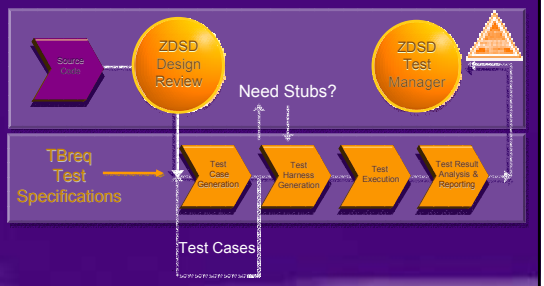
- Test Environments
- Host/Target Testing
- Integration with Test Verification
- Full Regression Testing Support

The Traditional Testing process

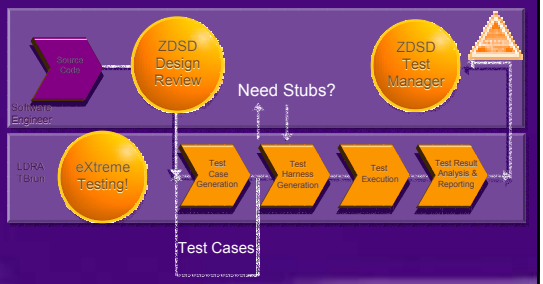


Labour Intensive
Limited Analysis & Understanding
Effect Driven
Regression Challenges

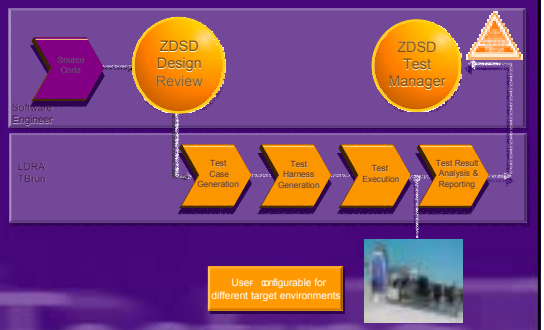
Requirements Validation Ecology



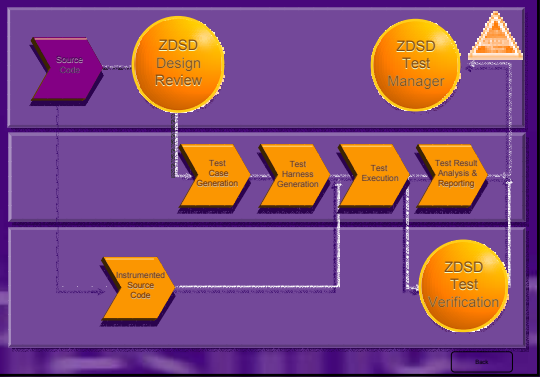
Design Verification Ecology



Unit Test – Automated Testing on Target

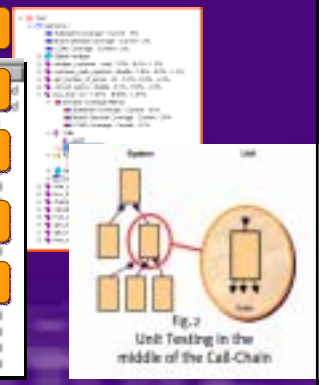


Unit Test – Integration with Test Verification



ユニットテスト自動化機能　：静的解析の情報を基に

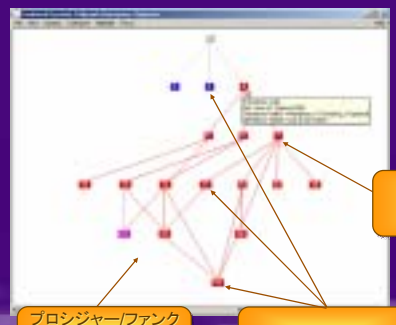
- テストハーネス自動生成
- スタブコードの自動生成
- グローバル変数の解決
- ソースの無いシステムコールもプロトタイプ化できる
- リグレッションテストをサポート



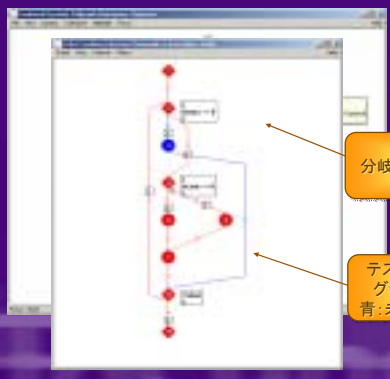
動的テストカバレッジ解析

- カバレッジ解析
- テストプランニング
- ダイナミックコールグラフ
- ダイナミックフローグラフ
- レポート表示
- データセットプロファイル

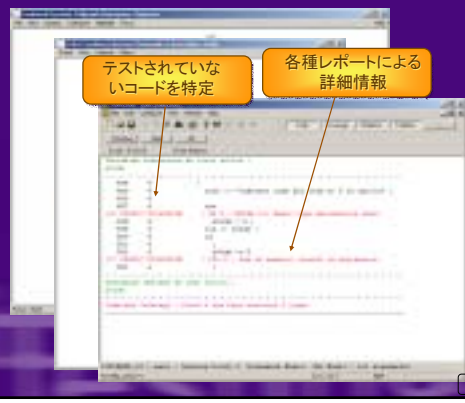
コールグラフ と テストカバレッジの融合



フローグラフ と テストカバレッジの融合



ソースコード表示 によるテストの評価



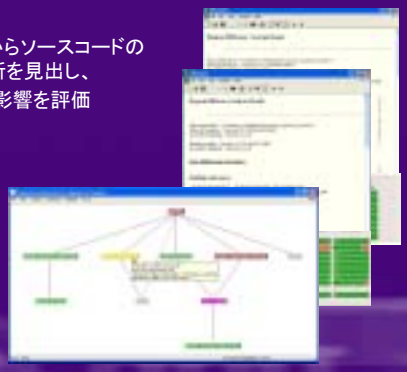
レポート機能 によるテストカバレッジの概要

テスト管理機能

- ソフトウエア変更箇所の検出
- リグレッションテストの管理
- テスト結果のドキュメント化

ソフトウエア変更箇所の検出

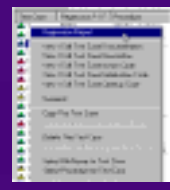
静的解析からソースコードの
変更箇所を見出し、
テストへの影響を評価



リグレッションテストの管理

テスト(スクリプト)とその結果は記録され、関連するコードやテストバージョンコントロールシステムと合わせて管理される

ユニットのリグレッションテストは、ビルドとテストの繰り返しサイクルの一環として自動化される

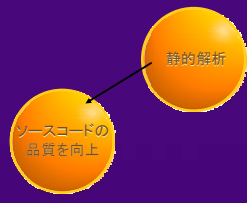


テスト結果のドキュメント化

完全なドキュメントによるレポートで、開発者、テスト担当者、保守、QA担当者全てに監査の為のデータを提供

レポートは、品質、メトリクスなどを高度に統合し概要を表示するが、各項目ごとへの詳細へ掘り下げた結果表示も可能

静的解析のみ？？ :従来のケース



静的解析,ユニットテスト,動カバレッジツールの融合

ソースコードに対するテストの深い洞察と、
開発サイクルにわたるテスト自動化を促進

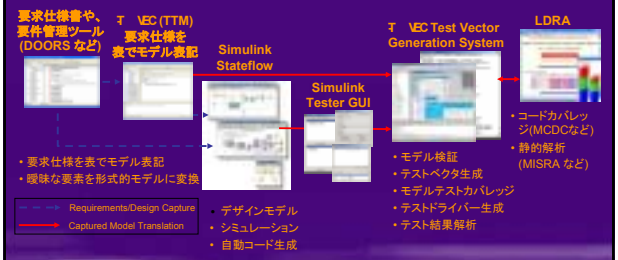
要求仕様、デザインモデル検証による
自動テスト生成・実行ツール

TVEC

We Cover All Boundaries

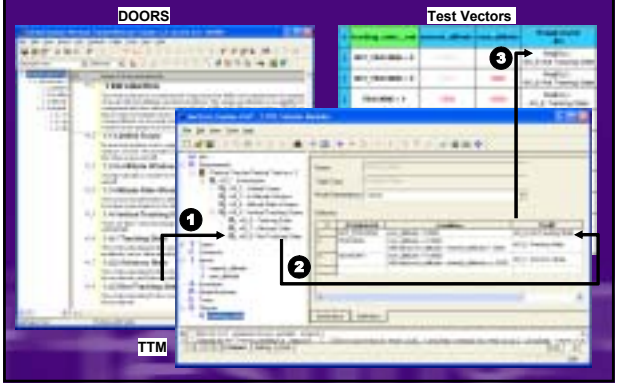
T-VEC Technologies, Inc.

TAF:テスト自動化のフレームワーク

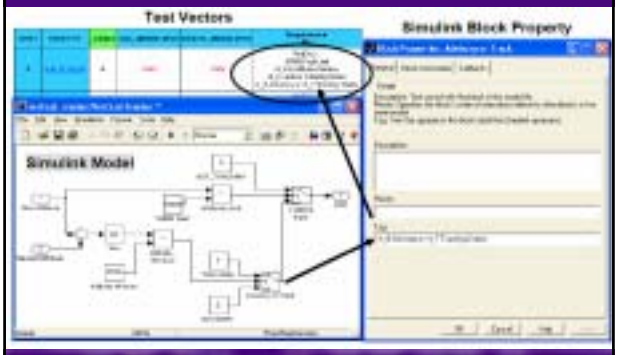


要求仕様～ Simulink モデル～テストへのトレーサビリティー
ロッキードマーチン社によるJSF開発事例

要求仕様とテストのリンク



Simulinkモデルとテストのリンク



モデルベース検証と自動テストの利点

- Model-Based Verification
  - **モデル検証**:
    モデル上の矛盾、不一致、ドメインエラー、ゼロ割りなどの欠陥を検出
  - **モデルベーステスト:（モデル検証の副次的な利点）**
    テストベクタ生成、モデルテストカバレッジ、テストドライバー生成、テスト実行、テスト結果解析、テストレポート生成

・テストケース生成、期待出力値の判定等のマニュアル作業を排除
・テストスクリプト（ホスト上、ターゲット上）の記述を自動化
・要求仕様やモデルに対する、テストカバレッジの解析
・モデルをバリデーションする為のシミュレーションデータの供給

Verification and Validation の自動化

- モデル検証
  モデル上の欠陥や、入力値範囲の見込み違いを発見
- モデル検証からテストを生成
  - モデル上の全論理パスに対してテスト入力と予測出力値のベクタ
  - テストシーケンスベクタは、マルチサイクルな実行にわたるテスト入力と予測出力値
- テストドライバーを生成
  - テストベクタやシーケンスベクタを基にテストドライバの自動生成
  - ソースコードへのテストドライバーの実行により、モデル対アプリケーションの検証
  - モデルの実証および、モデルに対するテストベクタ生成を検証するための、MATLABシミュレーション用テストドライバー
  - テストドライバーは如何なる環境へも対応

テストベクタ生成

- ユニット、インテグレーション、システムレベルテスト
- テストベクタは、テスト入力・期待出力値、及び
  モデルへのトレーサビリティを含む
- Simulinkモデルへの依存性なく、予測出力値が生成される
- テストカバレッジ、～MCDCレベル
  - Default - covers explicit paths (if, switch)
  - Path - covers all paths (implicit and explicit)
  - Condition - covers all conditions (MC/DC)
- モデル上の動的な振舞いの為の、テストシーケンスベクタ

テストベクタ生成（続き）

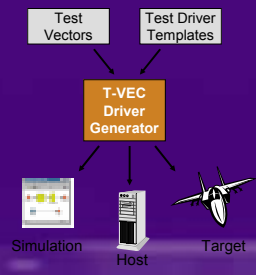
- モデル上の全コントロールパスを認識
- 各パスごとの収束条件が解析される
- テストは収束される領域に沿って選択される
- テストケースは入力領域に対するパス上の収束条件を加味して選択
- 追加の収束条件は、更なる特性の解析、実証など

テストドライバー生成

テストドライバー生成用テンプレートは、いかなる言語・スクリプト・実行環境に対応

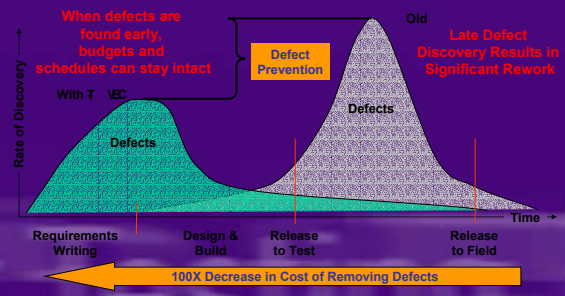
また、コードカバレッジツールなど他のツールとも柔軟に統合

- Included templates:
  - C test drivers for RTW GRT and ERT code generator
  - Simulation scripts for Matlab/Simulink



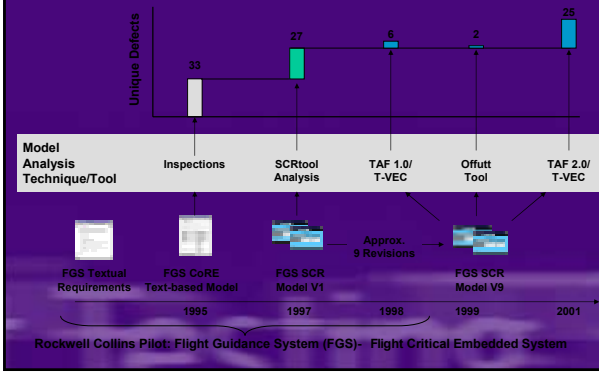
Early Defect Identification and Prevention

- Finding and fixing defects early produces costs savings



Source: Ed Safford, Lockheed Martin, Software Technology Conference, 2000.

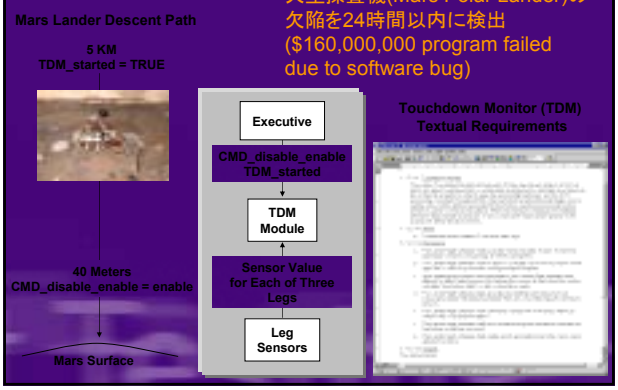
Model-based Analysis Finds Hidden Defects



Rockwell Collins Pilot: Flight Guidance System (FGS)- Flight Critical Embedded System

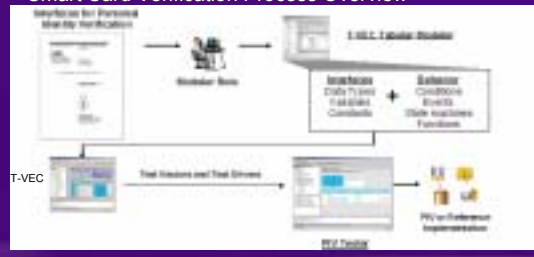
火星探査機における事例

火星探査機(Mars Polar Lander)の欠陥を24時間以内に検出
($160,000,000 program failed due to software bug)



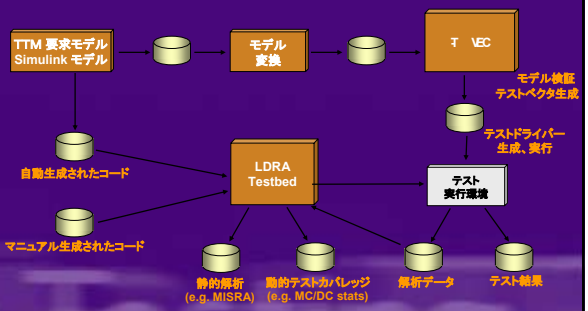
General Approach

- Smart Card Verification Process Overview



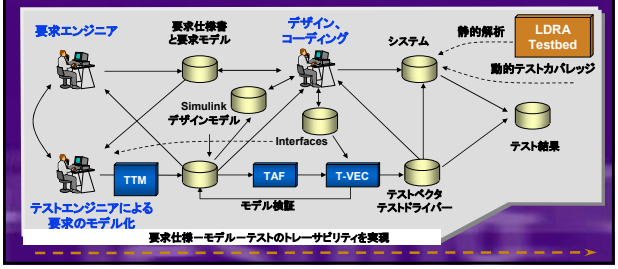
1. Building a SCR model specification for the PIV using TTM
2. Translating the TTM model into T-VEC specifications
3. Generating the test vectors from the T-VEC specifications
4. Generating test drivers to execute the test vectors against an implementation of the PIV
5. Execute the test driver (carried out by the PIV Tester)
6. Evaluating test results and produce test results report (carried out by the PIV Tester)

追加：LDRA社解析ツール



まとめ

Simulinkモデルや要求仕様モデルとは独立してモデル上の欠陥を解析し、テストベクタ　及びシーケンスベクタを自動生成。 生成されるテストドライバーはモデルが正しくコード化されたかの検証。 テストベクタはモデル上の各要素の検証、テストシーケンスベクタはモデル上の動的特性の検証に使用されうる。



ありがとうございました

Software Technology

We Cover All Boundaries



FUJI SETSUBI

富士設備工業株式会社　電子機器事業部
http://www.fuji-setsu.co.jp
E-Mail：info@fuji-setsu.co.jp