Characteristics of Software Defects

- Design Fault
  - Mostly introduced at the design stage
  - Deterministic, not random fault

Aspects for Improving Software Quality

- Development Process
- Correct Product
- Competent Organization

Purpose of S/W Qualification Activities

- Target System
  - Application Software for Digital Reactor Protection System (Safety-Critical Class)
  - Project Period: 2001 – 2008 (7 year project)

- Purposes of S/W Qualification Activities
  - Satisfaction of Licensing Criteria
    - Evaluation of Licensing Suitability
  - Improvement of S/W Quality
    - Finding S/W Defects at an Early Stage
    - Finding S/W Hazards along SDLC
    - Adherence of S/W Development Baseline
Main Activities of DRPS S/W Qualification

- Quality Assurance (QAM/QAP)
- S/W Qualification Activities
- S/W Safety Analysis
- S/W V&V
- S/W CM

Software Lifecycle Model

- SLCM: A Spiral Circulation of Waterfall Model
- SLC Phases: Phases in NUREG-0800 BTP/HICB-14
  - Planning, Requirements, Design, Implementation, Integration, Validation, Installation, Operation & Maintenance

Codes & Standards for DRPS S/W
S/W V&V Activities for DRPS Software

- Planning Phase
  - QA Plan
  - V&V Plan
  - CM Plan
  - Safety Plan

- Design Phase
  - Review of Plans
  - Traceability Evaluation
  - Test Preparation
  - Formal Verification

- Implementation Phase
  - Licensing Suitability
  - Core Inspection
  - Test (CT/IT/ST)

- Integration Phase
  - Licensing Suitability
  - Core Inspection
  - Test (CT/IT/ST)

- Validation Phase
  - Licensing Suitability
  - Core Inspection
  - Test (CT/IT/ST)

### Licensing Suitability Evaluation

- Purpose: Confirms whether or not the software requirements or design descriptions possess required software properties
- Checklist-Based Evaluation for Criteria of Software Properties (from NUREG-0800 BTP/HICB-14 & IEEE Std. 7-4.3.2)

<table>
<thead>
<tr>
<th>Item</th>
<th>R.G./Code/Std</th>
<th>Inspection Items</th>
<th>Inspection Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Completeness of S/W Functionality</td>
<td>BTP-HICB-14, Chap. 3.3a</td>
<td>Are all operational modes of the software defined?</td>
<td>Operation Bypasses: NOT complete in CPC-CWP</td>
</tr>
<tr>
<td>Safety</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Traceability</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Valuableity</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compressibility</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Usability</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Detailed Inspection (Fagan Inspection)

- Evaluating SRS/SDS with respect to Completeness, Consistency, Correctness
- A Checklist-Based Inspection for Functions, Inputs/Outputs, and Interfaces for Each of CCC.
Traceability Analysis

- **Scope and Type**
  - From system requirements to validation phase
  - Bidirectional (Forward/Backward) Traceability Analysis

![Diagram showing traceability analysis]

Formal Specification & Verification

- **Proprietary Tools of Formal Specification & Verification**
  - Requirements: NuSRS (based on NuSCR) + SMV Model Checking
  - Design: FBD Verifier

- **Application Scope**
  - S/W modules that take an important trip function
  - Formal verification tools usually cannot handle the entire S/W modules because of its capacity limitation of internal states

- **Merits & Demerits**
  - It can find a subtle S/W defect that cannot be indentified from document evaluation.
  - It is very difficult to use the tool and interpret its output.

- **Automatic Code Generation**
  - Necessary to reduce human coding errors ➔ but, bulky

Testing Activities on SWLC Phases
Software Safety Analysis

- Purposes of S/W V&V and S/W Safety Analysis (SSA)
  - S/W V&V: Functional/Process Requirements
  - SSA: Safety Requirements (To find out a potential hazard)
- SSA discovered defects that had not been identified in the V&V works → It can compensate for V&V works.
- It can indicate on which the testing resources will focus.

Activities of S/W Safety Analysis

- Activities
  - Identification of Software-Contributable System Hazards & Interface Points
  - Hazard Analysis of Process Characteristics
  - Hazard Analysis for Functional Characteristics
  - Examination of Organization & Responsibility
- Techniques for Functional Characteristics
  - Software HAZOP
  - Software FTA
  - Software FMEA

Software-Contributable System Hazards

- Software-Contributable System Hazards for DRPS App. S/W

<table>
<thead>
<tr>
<th>No</th>
<th>Hazard</th>
<th>Criticality Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DRPS cannot generate a trip signal when a trip condition for a process variable is satisfied.</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>DRPS generates a trip signal when it should not generate a trip signal.</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>DRPS cannot send qualified information of its operating status to the main control room.</td>
<td>2</td>
</tr>
</tbody>
</table>

Criticality Level 4 - The most significant hazard that can drive a plant to an accident.
Criticality Level 3 - A hazard that impacts significantly on the system operation but does not lead to an accident.
Criticality Level 2 - A hazard that can affect more or less the system operation.
Criticality Level 1 - An insignificant hazard that seldom affects the system availability.
Identification of Interface Points

- Interface Points between FBD Modules and Hazards
  - Trip modules in no.4 (except CPC_CWP) affect the hazard item 1 and 2.
  - Some S/W in FBD Module no.1 & 2 affect the hazard item 1 and 2 through Trip_Loc (no.4).
  - FBD modules of no.5, 8, & 9 affect hazard item 3.

Features of Technical Methods

<table>
<thead>
<tr>
<th>Software HAZOP</th>
<th>Software FTA</th>
<th>Software FMEA</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Documents + FBD Code</td>
<td>Defective S/W Module at Design/Implementation Phase</td>
<td>S/W Code (Integrated &amp; Issue)</td>
</tr>
<tr>
<td>All SW Contributable Hazards</td>
<td>Most Critical Hazard (CL-4)</td>
<td>All SW Contributable Hazards</td>
</tr>
<tr>
<td>Forward, Broad-Thinking Analysis</td>
<td>Backward, Local Systematic Analysis</td>
<td>Forward, Broad Systematic Analysis</td>
</tr>
<tr>
<td>Brainstorming by HAZOP Members</td>
<td>Fault Tree by an Individual Analyst</td>
<td>By an Individual Analyst</td>
</tr>
<tr>
<td>Deviation Quantity, Qualitative Functional Characteristics</td>
<td>Based on Fault Tree Templates for Function Blocks</td>
<td>Based on a Single Failure-Mode Template for All the Function Blocks</td>
</tr>
<tr>
<td>Guide Phrases (Rather Than Guide Words)</td>
<td>Logical Operation in Fault Event</td>
<td></td>
</tr>
<tr>
<td>Proven-Technology by NUREG/CR-6430</td>
<td>Addressed by NUREG/CR-6101 Appendix</td>
<td>Addressed Partly by NUREG/CR-6101 Appendix</td>
</tr>
<tr>
<td>Need Discussion Skills</td>
<td>Difficult to Apply to All Scope</td>
<td></td>
</tr>
</tbody>
</table>

Conclusions

- For safety critical software, the software V&V and safety analysis were applied to each phase of the software lifecycle.
- Both activities have different perspectives and compensate for each other (one for functional requirements, the other for safety requirements).
- Through the software V&V activities, numerous anomalies were identified at an initial lifecycle and these defects were resolved as the lifecycle proceeded.
- The software safety analysis could find different types of software defects that have not been identified in the V&V works.
- Software HAZOP/FTA/FMEA were all very useful, and S/W FMEA is better.
- Further Considerations
  - Cyber Security: How to incorporate it into V&V activities or be a separate process?
  - Quantification of S/W Quality and Reliability
One Further Activity: Quantification

- **Bayesian SRGM**
  - Purpose: Quantification of Software Code Quality after Testing
  - Identifying the degree of effectiveness of the current test and anticipating the amount of V&V testing in the next phase.

- **S/W Failure Probability Estimation**
  - Purpose: Reliability of Software
  - Quantification of V&V Results + Test Results.

- **Bayesian Belief Networks**
  - Purpose: Quantification of Quality of S/W Development Process
  - Quantification of All of V&V Activities

- **Difficulties in Applying Existing Reliability Model**
  - No Sufficient Failure Data to Apply to Existing S/W Reliability Models
  - Large Amount of Qualitative V&V Activities Revealing S/W Quality
  - Large Amount of Quantitative Test Activities...

Backup Materials

Formal Specification & Verification

- **NuSRS: Formal Requirements Specification/Verification Tool**
  - Formal Specification Language: NuSCR (A Modified SCR used in NPP)
  - NuSCR Language Types: Functional Overview Diagram (FOD), Structured Decision Table (STD), Finite State Machine (FSM), Timed Transition System (TTB)
**SRS Formal Verification by Model Checking**

- **Formal Verification**
  - Automatic Translation of NuSRS from NuSCR into SMV Input Format
  - Model Checking by Cadence SMV incorporated into NuSRS

- **Main Properties to be Verified**
  - Deadlock Freeness (System can never be in a situation where no progress is possible)
  - Non-Determinism (System never has some conflicting transitions)
  - Trip by Error (If module/channel/input error occur, trip signals are fired immediately)
  - Trip by Logic (Trip signal is fired if the process value rises above the setpoint and this condition lasts for pre-defined time)
  - Normal Status (If trip conditions are not satisfied, then trip signal shall never be fired)
  - Trip with OB (Trip signal is never be fired during operating bypass)

---

**FBD Formal Verifier**

- **Formal Verification Tool for Design and Implementation Phases**
- **Automatic Conversion of FBD Program into “VeeLog” Language**
- **Model Checking of Verilog Formal by SMV Model Checker**
- **The Same CTL Properties as That in NuSRS → Enable Consistency Check between Requirements and Design Phases**

---

**Safety Classification of I&C Systems**

- **Class-I**
  - Reactor Protection System
  - Safety Shutdown Instrument
  - Safety-Associated System
  - Safety Information System
  - Safety Data Center
  - Safety Quality and Security
  - Safety Manual Facility

- **Class-II**
  - Diverse Protection System
  - Diverse Manual Control
  - Diverse Actuation System

- **Class-III**
  - Plant Control System
  - Process Monitoring, Diagnosing, Processing, Alarm System
  - Fire Protection System
  - I&C in Radioactive & Spent Fuel

---

*Reference: KINS Reg. Guide, Chap. 8*
Target S/W: S/W in Digital RPS (IDiPS-RPS)

- Configuration of IDiPS-RPS (for each channel)
  - Two Bistable Processors (BPs): Determine trip state by signal comparison, SC
  - Two Coincidence Processors (CPs): Generate trip signal by a 2/4(2/3) voting, SC
  - Automatic Test & Interface Processor (ATIP): Performs Tests(MT/MIAT/PT) & Interfaces with other ATIPs, SR
  - Cabinet Operator Module (COM): GUI + H/W (Ch. Bypass, Init. Circuit Reset)
  - Network Type: SDL (SC), ICN (SR), ICDN

CASE Tool: SIS-RT

- SIS-RT: Proprietary Tool for KNICS Project
- Three View Modes: Inspection View, Traceability View, Structure View

Example of Formal Verification

- Accomplishment of Formal Verification
  - All Safety-Critical/Related Software Modules
  - Specification Errors (Vague Initializations, Multiple Test Case Selection), Behavioral Errors (OB)
**Example of Formal Verification**

- Accomplishment of Formal Verification
  - All Safety-Critical/Related Software Modules
  - Specification Errors (Vague Initializations, Multiple Test Case Selection), Behavioral Errors (OB)

**Test Coverage Criteria for FBD Program**

- Incorporate Test Coverage Criteria into Black Box Testing for Compensation of Each Other
- FBD program is converted into a corresponding flow graph and the test coverage criteria is applied to draw necessary test cases.

**Software Safety Analysis Activities**

- System Safety Analysis
  - Preliminary System Hazard Analysis
  - Identification of Software-Contributable System Hazard
  - Software Requirements HAZOP
  - Software Design HAZOP
  - Software Code HAZOP
  - Software Design FTA
  - Software Code FTA
  - Safety Checklist (for Interface, Constraints, Analysis Reports, Organization & Resources)
Analysis Method #1 - Software HAZOP

- **Distinguishing Features of Software HAZOP**
  - Deviation Quantity: S/W Functional Characteristics (Qualitative)
    - Accuracy, Capacity, Functionality, Reliability, Robustness, Security, Safety
  - Guide Phrases rather than Guidewords

<table>
<thead>
<tr>
<th>Phase</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theory</td>
<td>Theory</td>
<td>Example</td>
</tr>
<tr>
<td>Practice</td>
<td>Practice</td>
<td>Example</td>
</tr>
<tr>
<td>Software FMEA</td>
<td>Software FMEA</td>
<td>Example</td>
</tr>
<tr>
<td>Software FTA</td>
<td>Software FTA</td>
<td>Example</td>
</tr>
<tr>
<td>Software HAZOP</td>
<td>Software HAZOP</td>
<td>Example</td>
</tr>
</tbody>
</table>

Analysis Method #1 - Software HAZOP

- **Distinguishing Features of Software HAZOP**
  - Deviation Quantity: S/W Functional Characteristics (Qualitative)
    - Accuracy, Capacity, Functionality, Reliability, Robustness, Security, Safety
  - Guide Phrases rather than Guidewords
Analysis Method #2 - Software FTA

- Software FTA was applied to the critical FBD modules identified by the software HAZOP.
- Software FTA was constructed based on the fault-tree templates.
- There is a variety of fault-tree templates: Each function block in FBD program has its own fault-tree template.
- Software FTA provides a systematic way to find a local and logical software defect that is hard to find by S/W HAZOP or V&V works.

Fault Tree Template for OR Function Block

\[
\text{OR} = \text{IN}_1 \lor \text{IN}_2
\]

Analysis Method #3 - Software FMEA

- Software FMEA was performed based on the failure-mode template.
- There is a single failure-mode template that are equally applicable to all the function blocks of the FBD program code.
- Software FMEA provides a very systematic and efficient way to find a software defect that had been passed through tests.
**MT_STS_CHK Module for MT**

1) is for checking whether the TCP (Trip Channel Bypass) condition is satisfied.
2) is used to check whether the AB (All Trip-Channel Bypass) condition is satisfied (AB or TCP is active for MT actuation).
3) is for checking whether a processor to be tested is not in test prohibition mode (Prohibition should be 0).
4) is used to check whether any initiation circuit is not currently activated (IC should be not activated).

---

**“BYP_CHK” in BYP_CHK Module**

---

**Result of A Defect in MT_STS_CHK**

- **Consequences of a Defect in MT_STS_CHK Module**
  - When any other channel (say, Channel D) is already all bypassed, Channel A can execute the manual test even though its trip channel is not bypassed.

- **Difficulty in Finding Out This Defect through the System Test**
  - Four fully-interconnected RPS channels required.
  - More than at least 2 test persons.
  - Complex Test Procedures
    - Variables to be tested: 18 trip variables per channel
    - Channels to be tested: 4 channels
    - Test Type: Manual Test, Manual-Initiated Automatic Test