I&C for Plant Protection Systems (Software Quality Assurance)

2013 IAEA/ANSN ETTG Workshop on
Nuclear Safety Tailored for Regulators
(17~21 June, 2013)
Int. Nuclear Safety School, KINS, Korea

Name : Kim, Bok Ryul
E-mail : k060kbr@kins.re.kr

한국원자력안전기술원
KOREA INSTITUTE OF NUCLEAR SAFETY

CONTENTS

1. SOFTWARE QUALITY -
General Technical Background

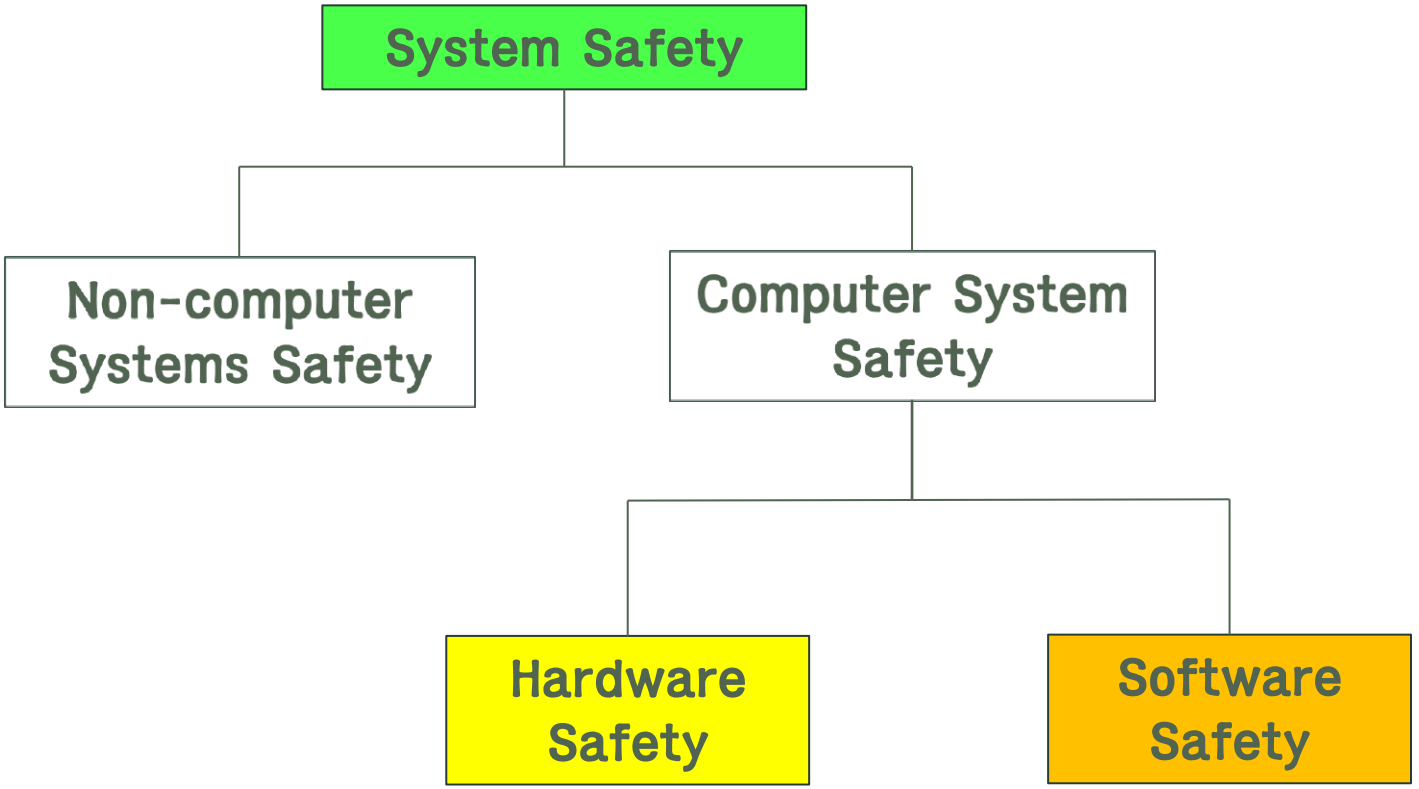
The Approach to Qualifying Software

- **Basis for Qualifying Software**
  - The Standard Review Plan uses a three step approach.
    - First, qualify the software development plans
    - Second, make sure the developer follows the plans
    - Third, make sure the outputs of the work are acceptable

Review emphasizes design quality, defense-in-depth, and diversity

Software Safety (1/6)

Software Safety Context



Software Safety (2/6)

- Roles of software that may have in safety applications
  - Software faults may contribute to an accident.
  - Software may be used to move a system from a hazardous state to a non-hazardous state.
  - Software may be used to help mitigate the consequences of an accident.

Software Safety (3/6)

- **Software failures that can lead to an accident.**
  - Inadvertent response to stimuli
    - An unexpected or unwanted event occurs
  - Failure to response when required
    - A planned event does not occur.
  - Response out of sequence
    - A known and planned event occurs but not when desired.
  - Response in unplanned combination with other software or other system components
    - Two or more portions of the system attempt to initiate inconsistent events
  - Improper magnitude or direction of response
    - This usually indicates an algorithm error.

Software Safety (4/6)

Typical Software Failure Modes

- Given the correct inputs, the program produces the wrong outputs.
- Given correct and timely inputs, the program produces outputs too late.
- Given correct inputs, the program does not respond.
- Given incorrect inputs, the program does not detect this.
- In the face of equipment malfunctions or design change, program actions are incorrect.

Software Safety (5/6)

- **Factors that may affect software safety**
  - Software reliability
  - Software performance
  - Development methods, practices, and procedures
  - Interconnections between software and the rest of the computer system
  - Interconnection between software and the rest of the application system, etc.

Software Safety (6/6)

- **Types of Software Common Mode Failure (CMF)**
  - Mistakes in a single set of requirements
  - Faults in a single design used for coding
  - Use of common code which contains faults in parallel paths
  - Use of common compilers and other tools which may contain faults
  - Common configuration management errors

SRP Review Process for Digital Systems (1/3)

Review Process Topics (SRP 7.0-A)

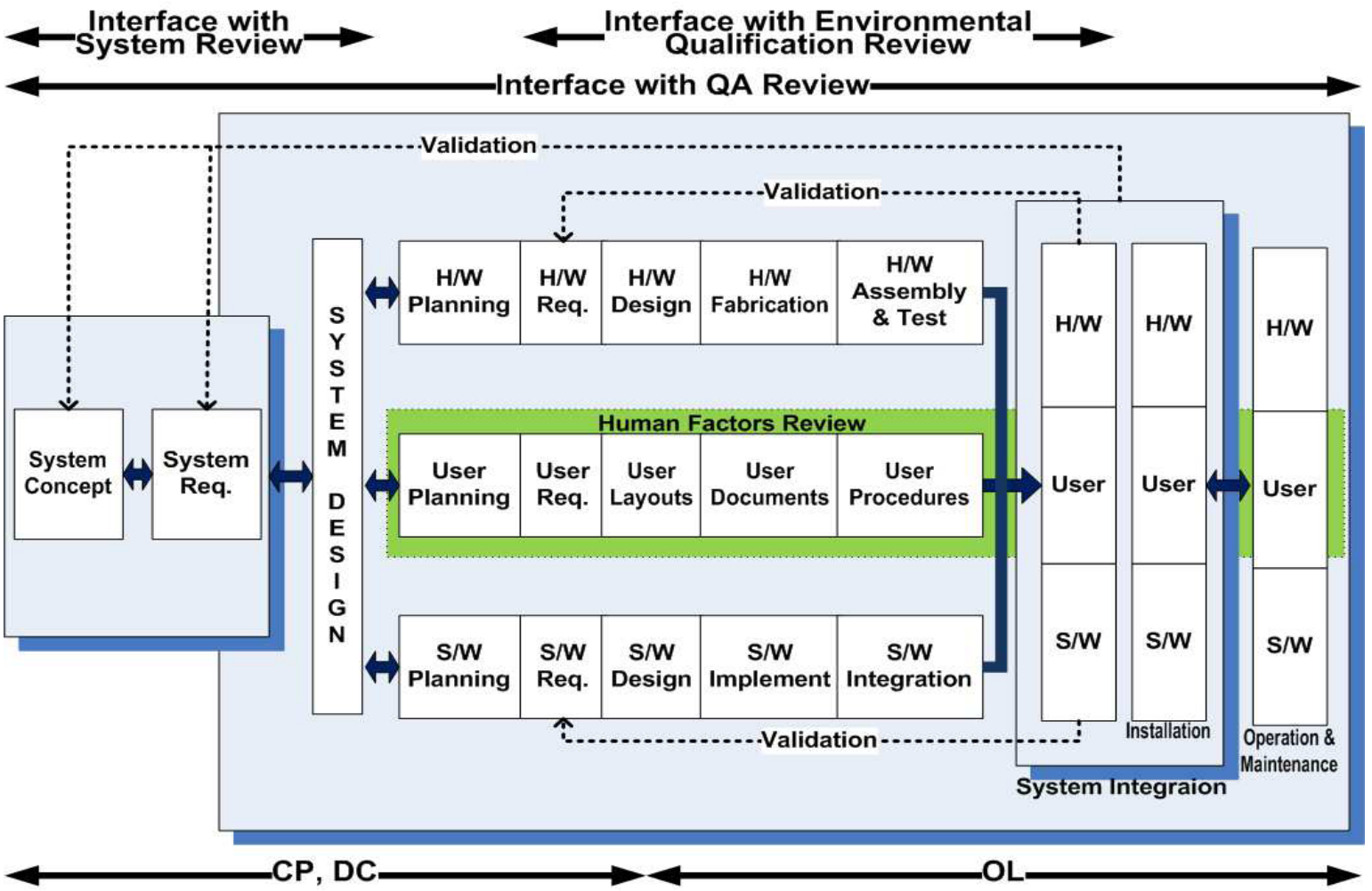
1. The adequacy of design criteria and guidance to be applied to the proposed system.
2. Identification of review topics – The subsequent review process depends upon the I&C systems addressed in the application.
3. Defense-in-depth and diversity – For applications that involve a Reactor Trip System or an Engineered Safety Features Actuation System, the ability of the combination of I&C systems to cope with common mode failure should be reviewed.
4. Life cycle process planning – The adequacy of the computer system development process, particularly the software life cycle activities for digital systems, should be reviewed.

SRP Review Process for Digital Systems (2/3)

Review Process Topics (SRP 7.0-A)

5. The adequacy of system functional requirements and commitments for the individual I&C systems – The requirements for each system are outlined in Sections 7.1 through 7.9.
6. The adequacy of the software life cycle process implementation – A sample of verification and validation, safety analysis, and configuration management documentation for various life-cycle phases should be audited.
7. Software life cycle process design outputs – The conformance of the hardware and software to the functional and process requirements derived from the design bases should be audited.

SRP Review Process for Digital Systems (3/3)



2. SOFTWARE QUALITY -
Planning Activities

Overview (1/2)

Context for Software Planning

- Planning the software effort should occur in the context of the overall application system planning
- Planning of safety-critical software in general should occur within the context of the system hazard analysis, system risk analysis, and system design
- This implies that the software plans may be components of more general instrumentation plans, or even project plans.
  - For example, the software safety plan might be a portion of the system safety plan.

Overview (2/2)

General Assessment Criteria

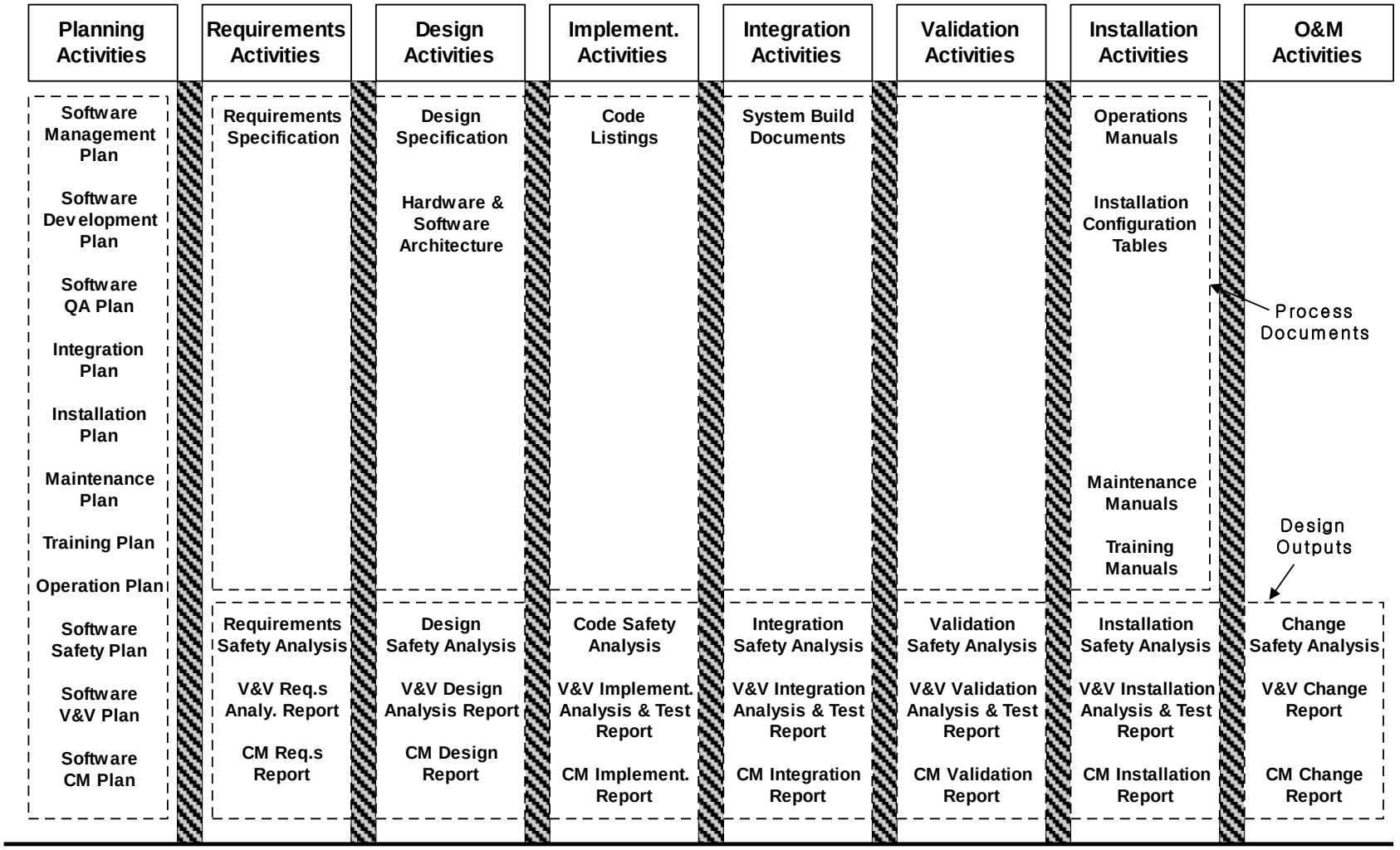
- Each plan should be internally consistent.
- The complete set of plans should be mutually consistent.
- Plans should be documented so they can be understood by users of the plan and regulatory reviewers.
- The set of plans should not be ambiguous.
- It should be possible to verify that the plans have been followed during the software project.

Review and assessment of the quality of the plans provides a means of judging the competency of the development organization and its management.

Software Life Cycle Process Planning (1/2)

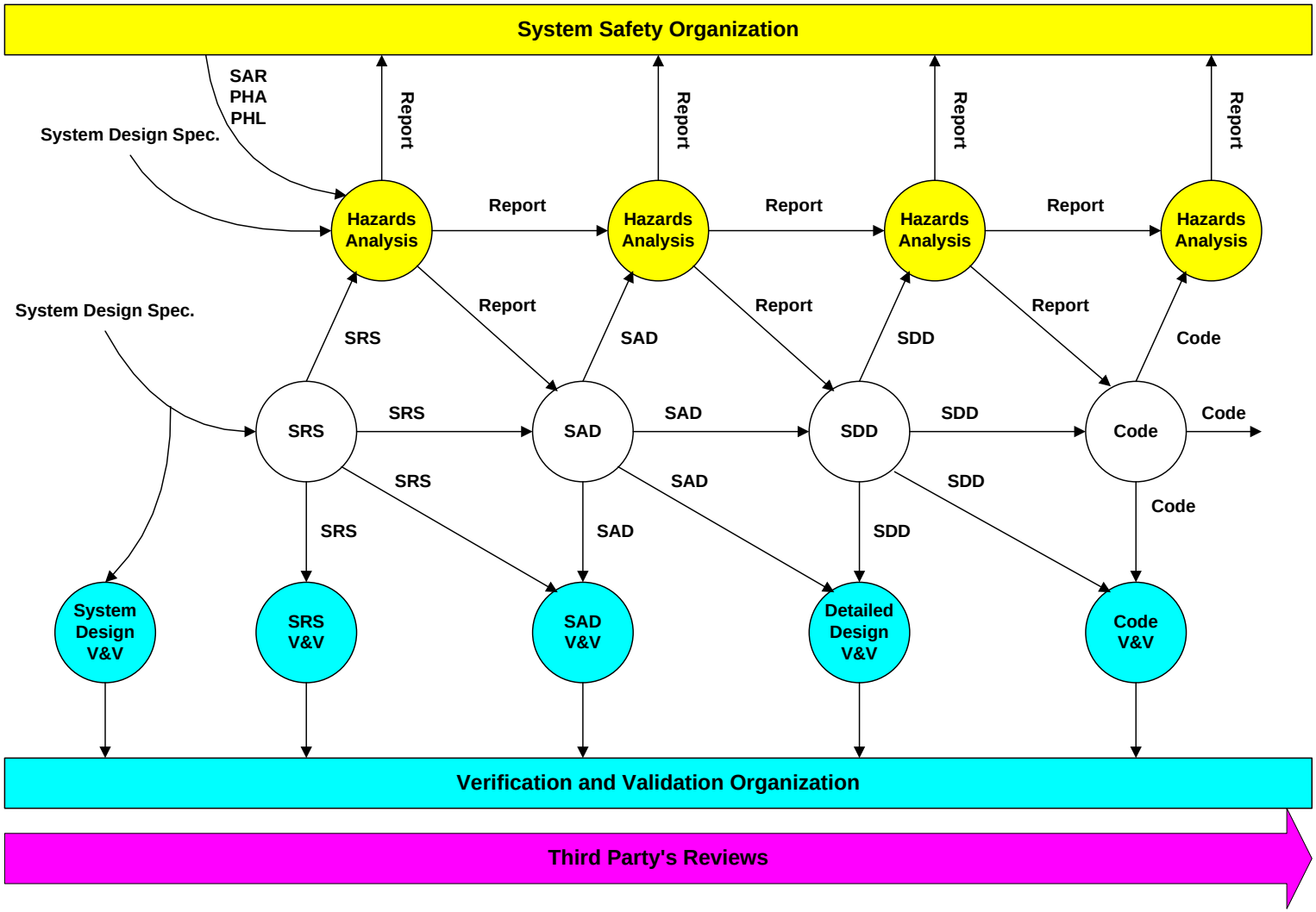
- Software Management Plan
- Software Development Plan
- Software Quality Assurance Plan
- Software Integration Plan
- Software Installation Plan
- Software Maintenance Plan
- Software Training Plan
- Software Operations Plan
- Software Safety Plan
- Software Verification and Validation Plan
- Software Configuration Management Plan

Software Life Cycle Process Planning (2/2)



3. SOFTWARE QUALITY -
Software Life Cycle Implementation

Software QA Activities for Safety-Critical Software



Software Safety Analysis Activities (1/2)

- **IEEE Std. 1228 – Software Safety Plan**
  - Presents a philosophy of software safety planning
    - Embeds software system in system safety
    - Software can have a negative or positive effect on safety.
    - Life cycle based from initial plans to final retirement
  - Describes a software safety plan
    - Management actions
    - Technical safety analyses
    - Post-development actions

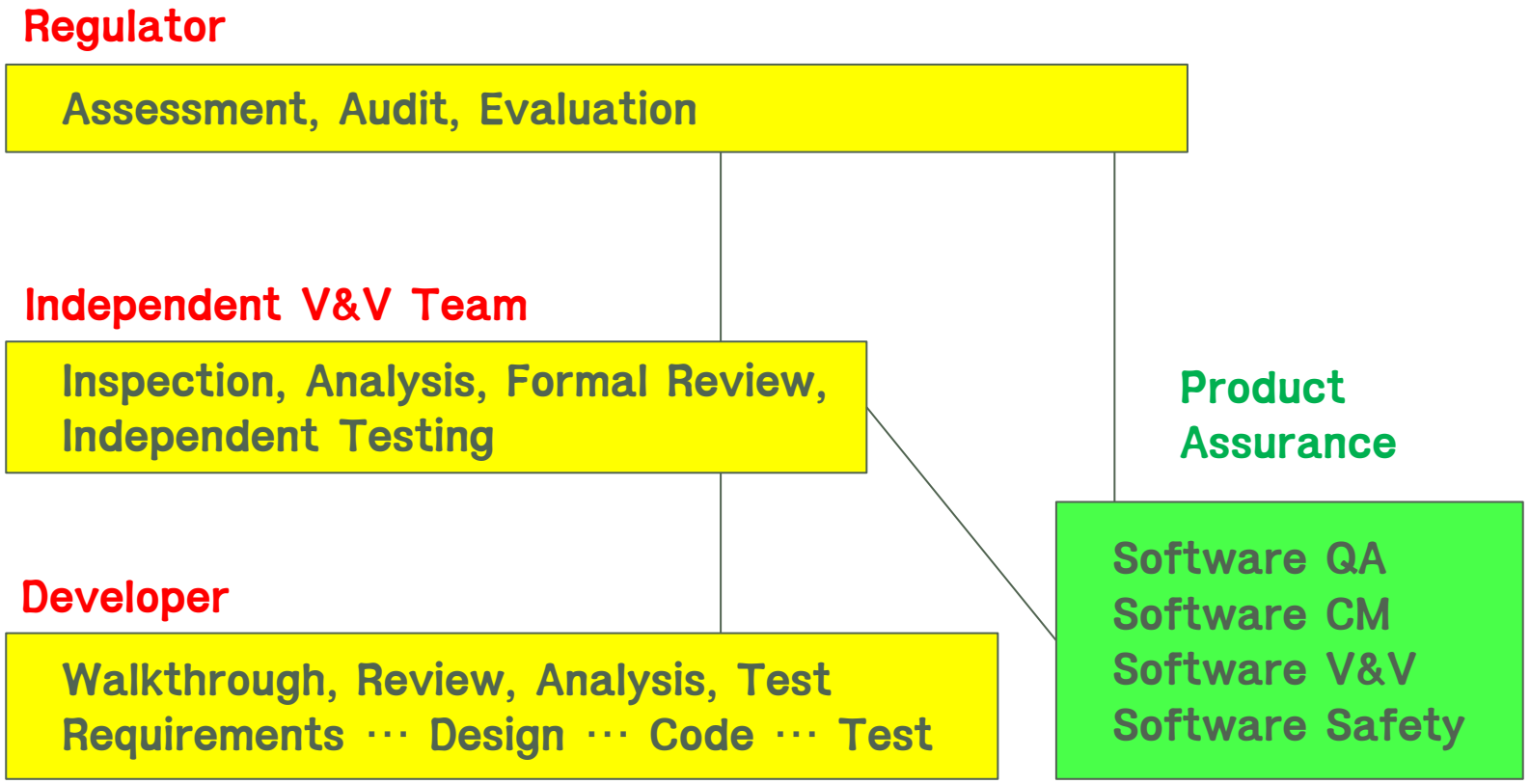
Software Safety Analysis Activities (2/2)

Software Analysis Activities

- Carried out for each group of life cycle activities
- Demonstrate that the Plan has been carried out
- Show that the system safety requirements have been adequately addressed for each activity group
- Show that no new hazards have been introduced
- Show that the software design outputs (requirements, design, code) that can affect safety have been identified.
- Show that all other software requirements, design elements and code elements will not adversely affect safety.

Software V&V Activities (1/5)

Perspectives on Verification and Validation



Software V&V Activities (2/5)

Specific Goals of Software V&V

- Detect and correct defeats as early as possible in safety critical software
- Lessen the chances of cost and schedule overruns for safety critical software
- Enhance safety critical software for quality and reliability
- Assess the consequences of proposed changes to software components
- Improve regulator visibility into the software development process for safety critical software

Software V&V Activities (3/5)

Degree of Independence

- IEEE Std. 1012 defines independence in terms of three parameters
  - Technical independence
  - Managerial independence
  - Financial independence
- The degree of independence of the V&V effort is defined by the extent each of the three independence parameters is vested in the V&V organization.
- The ideal independent verification and validation (IV&V) contains all three independence parameters.

Software V&V Activities (4/5)

Forms of IV&V

IV&V Form	Technical	management	Financial
Classical	I	I	I
Modified	I	i	I
Integrated	i	I	I
Internal	i	i	i
Embedded	e	e	e
Note: I = rigorous independence; i = conditional independence; e = minimum independence			

Software V&V Activities (5/5)

General Assessment Activities

- Verify that the tasks described in the V&V plan have been implemented completely.
- Documentation should exist to demonstrate the successful complete of each V&V task.
- Problems identified by verification efforts should be documented, and a record should be kept of actions taken in response to the problems.
- A traceability matrix should be produced showing the linkage from system requirements imposed on software, through software requirements, software design, code, integration, validation and installation.

Software CM Activities (1/3)

Overview

- Reviewer verifies that the configuration management (CM) plan has been followed.
- Configuration baselines should exist for the activity group.
  - Defines the basis for further development
  - Allows control of configuration items
  - Permits traceability between configuration items
- Control and document changes to the baseline
  - Configuration control board should exist.

Software CM Activities (2/3)

Configuration Items

- Software requirements
- Software designs
- Code
- Support software used in development
- Libraries of software components essential to safety
- Software plans that could affect safety
- Test plans, specifications, procedures, cases, results
- Software documentation
- Databases and software configuration data
- Pre-developed software items
- Software Change documentation
- Tools used in project management, development, assurance

Software CM Activities (3/3)

Benefits of Software Configuration Management

- Control and manage software products throughout their life cycle
- Reduce effort to identify and implement changes
- Ensure only necessary changes will be made
- Extend the useful life of software products
- Reduce defects in software releases
- Improve customer satisfaction
- Provide a natural feedback mechanism

Software configuration management enables one to know which software products are being verified and validated.

4. SOFTWARE QUALITY – Digital Plant Protection Systems Used in Uljin Units 5 and 6

Software Classification (1/2)

Software Classification

- The software of the DPPS is identified by designers as the following classes:
  - **Protection**: Software whose function is necessary to directly perform RPS, ESFAS protective actions and safe shutdown actions.
  - **Important to Safety**: Software whose function is relied on to monitor or test protection functions(ITP & MTP), or software that monitors plant critical safety functions.
  - **General Purpose**: Software that performs some purpose other than that described in the previous categories. This includes software tools that are used to develop software in the other category, but is not installed in the on-line plant system.

Software Classification (2/2)

Assignment of DPPS Software to Classes

SYSTEM	SUB-SYSTEM SCOPE	CLASS
DPPS	Safety Critical Kernel (LCL, Bistable, CCC)	Protection
	MTP Test Processor Software	Important to Safety
	ITP Test Software	Important to Safety
	Inter-System Communication Software	Important to Safety
	All Other Software	General Purpose
	ADVABUILD	General Purpose

Software V&V and Development Process (1/2)

Software Life Cycle and Code Verification Methods

Software V&V Phases	Verification Methods
Concept	- Traceability - Platform evaluation reviews
Requirements	- Traceability
Design	- Traceability - Commercial dedication plan review
Implementation and Coding	- Source code traceability - Code Inspection for technical adequacy, design integrity, completeness of interfaces - In-process audits for adherence to coding standards and configuration controls
Testing	- Unit test procedures and test reports - Module test procedures and test reports

Software V&V and Development Process (2/2)

Software Life Cycle and Code Verification Methods

Software V&V Phases	Verification Methods
Testing	- Software integration procedures and test reports - Hardware verification procedures and test reports - Factory acceptance procedures and Test reports - Integrated System validation procedures and test reports - Inter-system procedures and test reports
Installation and Check out	- Test procedures and test reports - Audits
Operations and Maintenance	- Audits
Retirement	- Not Applicable

Software V&V Tasks and Responsibilities (1/3)

TASK	PROTECTION	IMPORTANT TO SAFETY	GENERAL
SQA Planning Phase			
- System Specific SVVPs	VT/DT	VT/DT	VT/DT
- System Specific SCMPs	DT/VT	DT/VT	DT/VT
Software Requirements Phase			
- System Design Specification	By others	By others	By others
- Functional Requirements	DT/VT	DT/VT	DT
- Software Requirement Specification	DT/VT	DT/VT	DT/VT
Software Design Phase			
- Software Design Description	DT/VT	DT/VT	DT
- Prototype Coding	DT	DT	DT
Software Implementation Phase			
- Test Plan (Part of SVVP)	VT/DT	VT/DT	DT
- Unit Coding	DT/VT	DT/VT	DT
- Unit Test Procedure	VT/DT	NA	NA
- Unit Test Execution	VT	NA	NA
- Unit Test Results Report	VT/DT	NA	NA

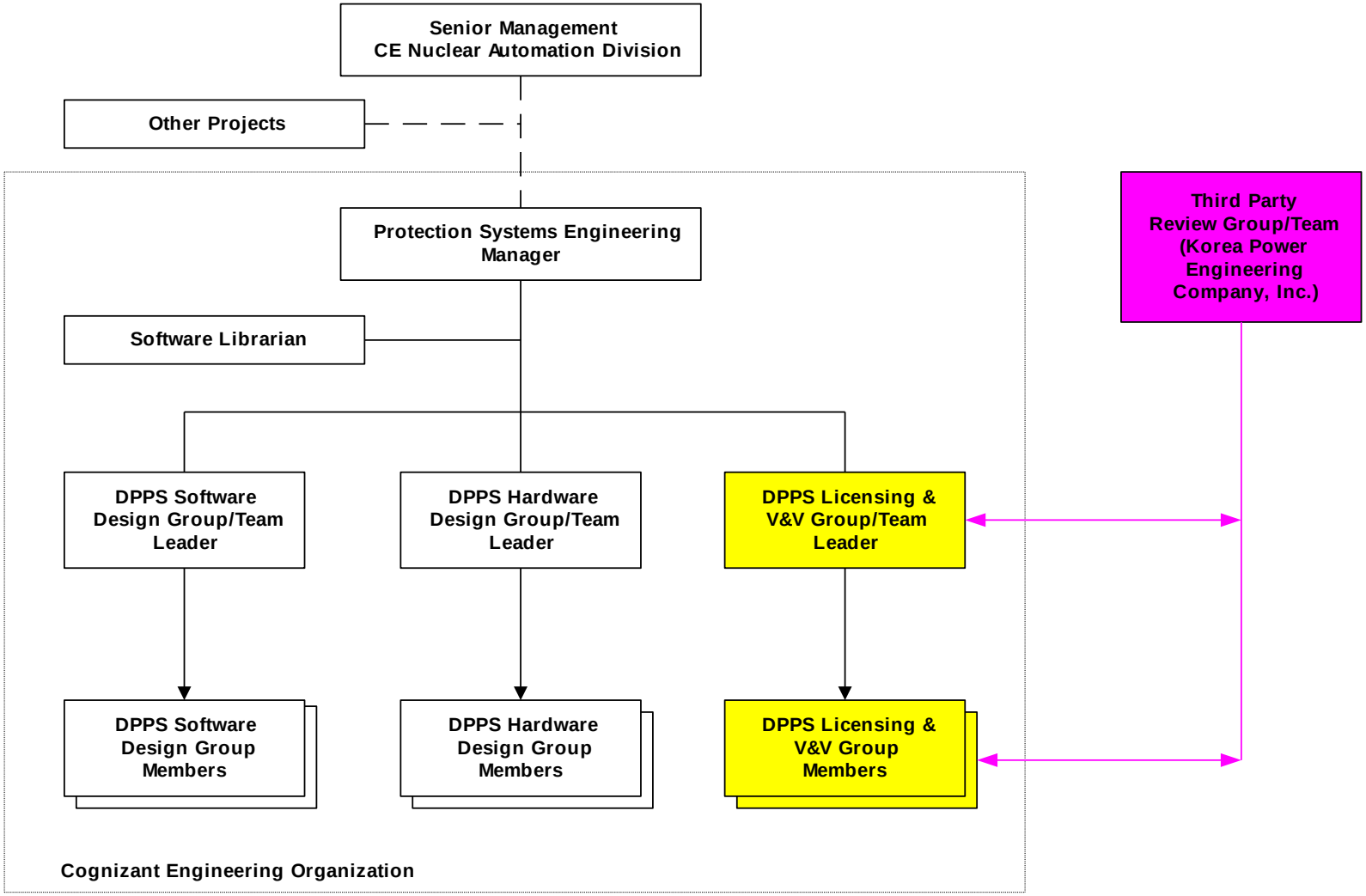
Software V&V Tasks and Responsibilities (2/3)

TASK	PROTECTION	IMPORTANT TO SAFETY	GENERAL
Software Implementation Phase			
- Module Test Procedure	VT/DT	DT/VT	NA
- Module Test Execution	VT	DT	DT
- Module Test Results Report	VT/DT	DT/VT	DT/VT
- Software Integration Test Procedure	DT/VT	DT/VT	DT
- Software Integration Test Execution	DT	DT	DT
- Software Integration Test Report	DT/VT	DT/VT	DT
Testing Phase			
- Integrated System Validation Test Procedure	VT/DT	VT/DT	VT/DT
- Integrated System Validation Test Execution	VT	VT	VT
- Integrated System Validation Test Results Report	VT/DT	VT/DT	VT/DT
- Factory Acceptance Test Procedure	DT/VT	DT/VT	DT/VT
- Factory Acceptance Test Execution	DT/VT	DT/VT	DT/VT

Software V&V Tasks and Responsibilities (3/3)

TASK	PROTECTION	IMPORTANT TO SAFETY	GENERAL
Testing Phase			
- Factory Acceptance Test Report	DT/VT	DT/VT	DT/VT
- Inter-System Test Procedure	DT/VT	DT/VT	DT
- Inter-System Test Execution	DT	DT	DT
- Inter-System Results Report	DT/VT	DT/VT	DT
- User Documentation	DT/VT	DT/VT	DT/VT
- Software V&V Report	VT/DT	VT/DT	VT/DT
- Software Acceptance Test Procedure	DT/VT	DT/VT	DT/VT
- Software Acceptance Test Report	VT	VT	VT
	DT = Design Team, VT = V&V Team		

Independent V&V Teams



Always we keep watching our Atomic Power

Name : Kim, Bok Ryul
E-mail : k060kbr@kins.re.kr

Thank You

한국원자력안전기술원
KOREA INSTITUTE OF NUCLEAR SAFETY