
SOFTWARE
QUALITY
MANAGEMENT
ANGÉLICA DE ANTONIO

COURSE OUTLINE

- PART 1: INTRODUCTION TO SOFTWARE QUALITY
- PART 2: SOFTWARE QUALITY CONTROL ACTIVITIES
- PART 3: QUALITY METRICS
- PART 4: QUALITY MANAGEMENT AND QUALITY SYSTEMS
- PART 5: SOFTWARE QUALITY ASSURANCE ACTIVITIES

- PART 6: SOFTWARE CONFIGURATION MANAGEMENT

SOFTWARE CONFIGURATION MANAGEMENT

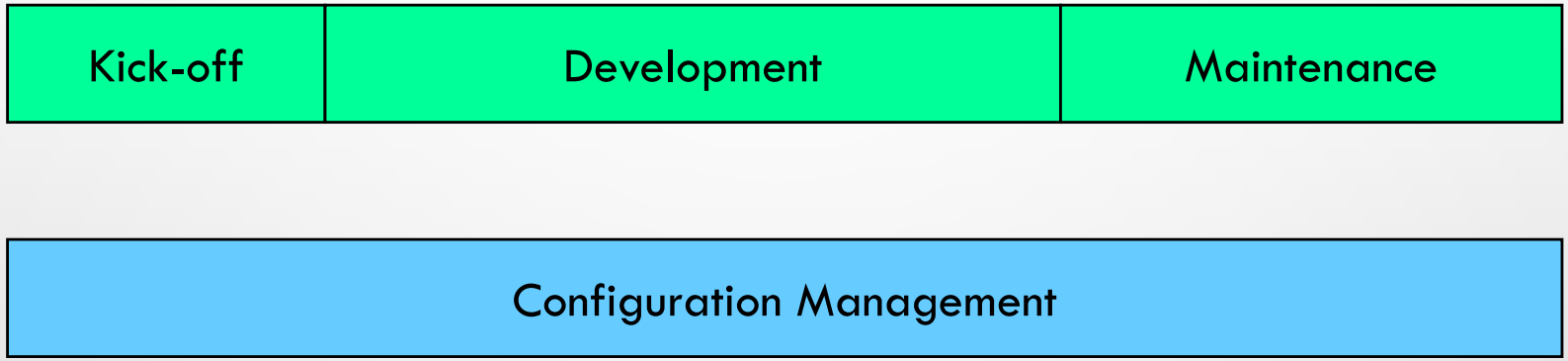
- Basic concepts of SCM
- Configuration Identification
- Configuration Change Control
- Configuration Audit
- Configuration Status Accounting
- Configuration Management Plan

WHAT IS SCM

- *ON ANY TEAM PROJECT, A CERTAIN DEGREE OF CONFUSION IS INEVITABLE. THE GOAL IS TO MINIMIZE THIS CONFUSION SO THAT MORE WORK CAN GET DONE. THE ART OF COORDINATING SOFTWARE DEVELOPMENT TO MINIMIZE THIS PARTICULAR TYPE OF CONFUSION IS CALLED CONFIGURATION MANAGEMENT.*
- *CONFIGURATION MANAGEMENT IS THE ART OF IDENTIFYING, ORGANIZING, AND CONTROLLING MODIFICATIONS TO THE SOFTWARE BEING BUILT BY A PROGRAMMING TEAM. THE GOAL IS TO MAXIMIZE PRODUCTIVITY BY MINIMIZING MISTAKES.* (BABICH)

WHEN TO MAKE SCM

- THROUGHOUT ALL LIFE-CYCLE



KINDS OF SOFTWARE EVOLUTION

- ASSOCIATED TO PRODUCT DEVELOPMENT
- ASSOCIATED TO DEFECTS OR DEVIATIONS WITH RESPECT TO REQUIREMENTS
- ASSOCIATED TO CHANGES IN THE REQUIREMENTS

THE CHANGE

- CANNOT BE AVOIDED!
- IT IS JUSTIFIED: AS TIME GOES BY, MORE IS KNOWN ABOUT
  - THE PROBLEM
  - HOW TO SOLVE IT

GOALS OF SCM

- PROVIDE VISIBILITY
  - OVER THE STATE OF THE PRODUCT AT ALL TIMES
  - OVER ITS EVOLUTION
- MAINTAIN THE INTEGRITY OF THE PRODUCT
  - WITH RESPECT TO REQUIREMENTS

SCM ACTIVITIES

Configuration Identification

- Visibility over the product
  - Components
  - Structure

Configuration Change Control

- Control of the Evolution

Configuration Audits

- Integrity Checking

Status Accounting

- Reports, records, statistics

DEFINITIONS

Configuration

- Any product used or generated in a software project

Software Configuration Item (SCI)

- Component of the configuration
- Working unit for SCM

SOFTWARE CONFIGURATION ITEMS

Possible Items

- Documents
- Programs
- Files
- Data Bases
- etc.

To be decided for each project

- What elements will be SCI
- Which ones will be “under configuration control”
- Decomposition of the SCIs
  - Size
  - Complexity
  - Need of visibility and control

BASELINE

Concept to facilitate change control

Selected milestones at the end of certain phases

- Identification of the phase's results
  - Approval of one or more SCI, by a Formal Technical Review (Inspection)
- Certify the end of the phase
- Set a solid base for future developments
  - They can only be changed by a "formal" change control process

COMPLEXITY OF SCM

Number of SCIs to control

- Grows as development progresses
- Depending on the product's size

Number of changes: lets try to minimize them

- Changes in requirements
  - Good Requirements Management
  - SOMETIMES UNAVOIDABLE: As soon as possible
- Defects
  - Good Quality Management

NECESITY OF SCM

Big systems

- Visibility and control over a high number of components is more difficult

Complex systems

- The more relationships among components the more difficult to evaluate change impact
- More difficult visibility

Long life systems

- People move and take what they have in their memory with them
- Long evolution paths with multiple versions

NECESITY OF SCM (II)

Big organization

- Each person does only know their portion of the product
- Problems with coordination of modifications

Need to maintain multiple simultaneous versions

- Which version(s) is(are) affected by a change

VERSION CONTROL

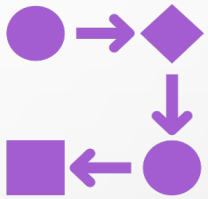
- FACILITATES SCM
  - KNOWING FOR EACH SCI:
    - THE LATEST VERSION
    - RELATIONSHIP AMONG VERSIONS (EVOLUTION)
    - LOCATION OF VERSIONS
  - PROVIDES ANSWERS TO CHANGE CONTROL
    - WHICH VERSION(S) ARE AFFECTED BY A CHANGE

DEFINITIONS

VERSION

Instance of a SCI

- at a given time during development
- stored in a repository
- that can be recovered at any time for its use or modification

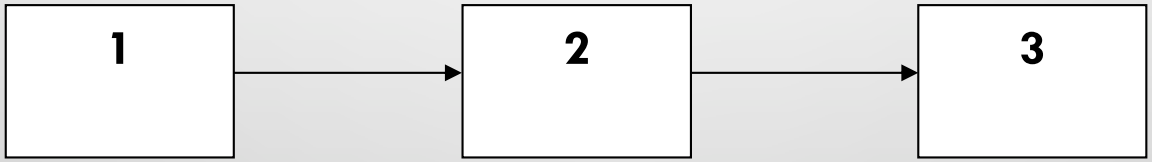


REVISION

Succesive versions that appear along time, as the development of a SCI evolves

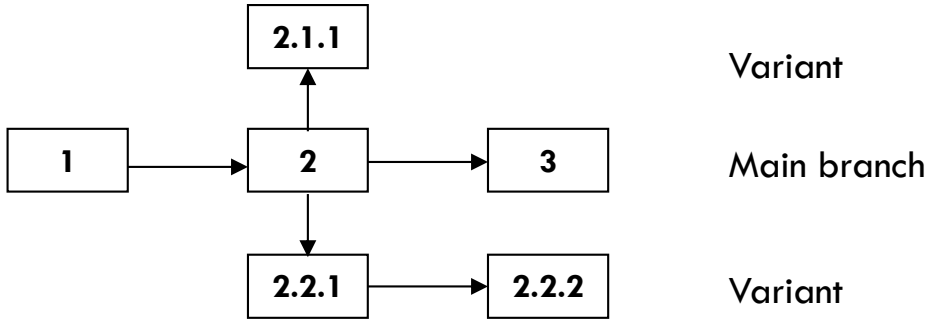
IDENTIFICATION AND RELATIONSHIPS AMONG REVISIONS

- EACH REVISION MUST BE UNIQUELY IDENTIFIED
  - USUALLY NUMERICAL SCHEMAS
- EVOLUTION GRAPH : REPRESENTATION OF THE DIFFERENT REVISIONS OF A SCI AND THEIR SUCCESION RELATIONSHIPS



VARIANT

- VERSIONS OF A SCI THAT
  - COEXIST AT A GIVEN TIME
  - DIFFER IN CERTAIN CHARACTERISTICS
- REPRESENTING THE NEED FOR AN OBJECT TO SATISFY OPPOSING REQUIREMENTS SIMULTANEOUSLY
- A VARIANT DOES NOT REPLACE ANOTHER VERSION (LIKE REVISIONS), BUT THEY OPEN UP NEW DEVELOPMENT PATHS (BRANCHES IN THE EVOLUTION GRAPH)



TYPES OF VARIANTS

TEMPORAL

- They will be merged with other variant some time ahead during development
- When several people are working simultaneously on the same version. A variant is created for each one to allow parallel work without conflicts

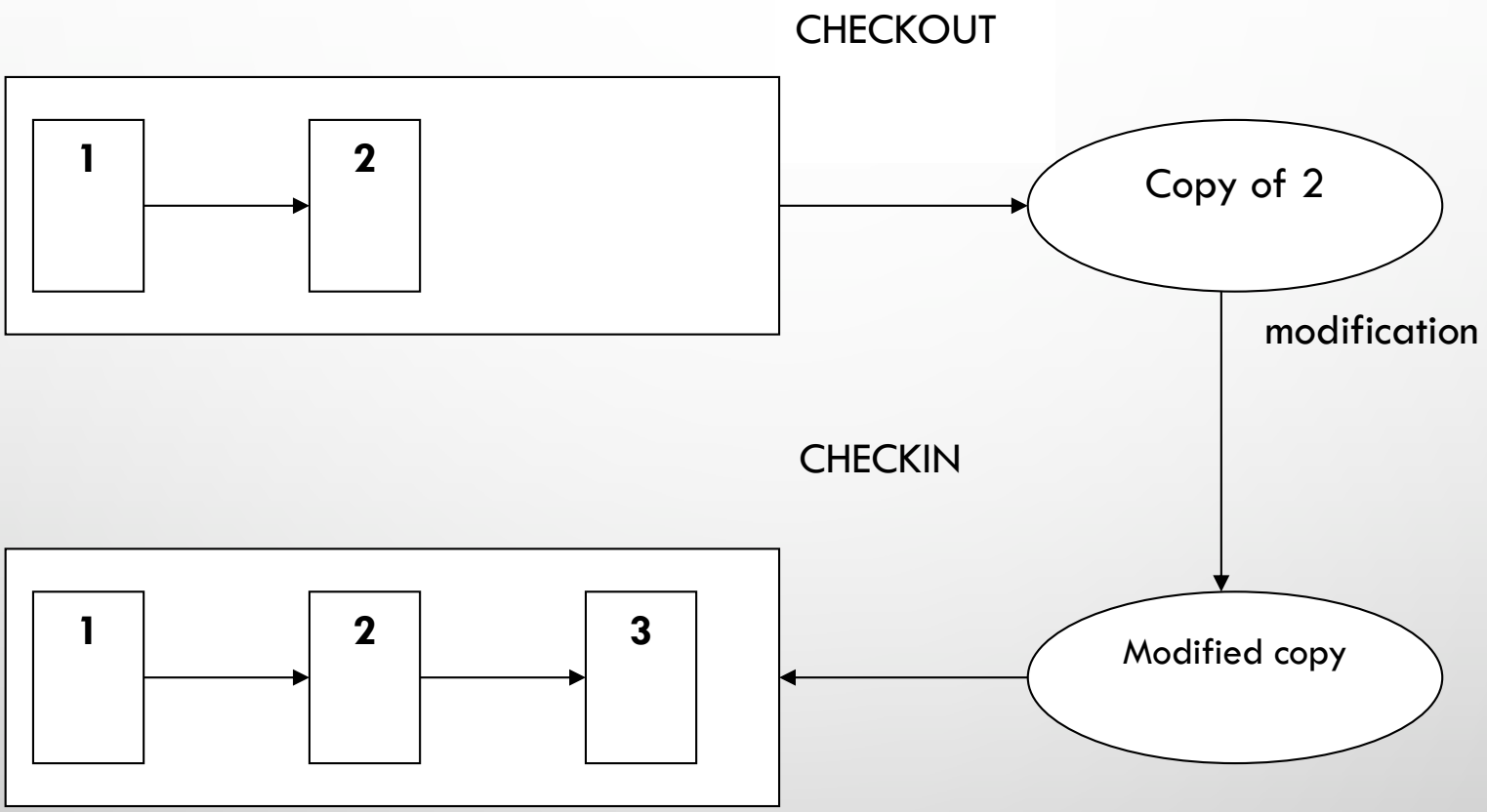
DISPOSABLE

- To explore alternative solutions and select the best one
- Test variants - with additional code to facilitate testing

PERMANENT

- User requirements variants - different languages typically
- Platform variants - one for each operating system or hardware platform

WORK MODEL


CHECKOUT
1
2
Copy of 2
modification
CHECKIN
1
2
3
Modified copy

STRATEGIES FOR CREATING REVISIONS

- EVERY MODIFICATION LEADS TO THE CREATION OF A NEW REVISION
  - PROBLEM: HIGH NUMBER OF REVISIONS, UNMANAGEABLE
- THE USER DECIDES WHEN A NEW VERSION MUST BE CREATED
  - THE OBJECT CAN BE MODIFIED AS MANY TIMES AS DESIRED, MAINTAINING THE SAME VERSION IDENTIFICATION
  - THE USER DECIDES WHEN THE VERSION MUST BE “FROZEN” - THE VERSION IS STORED AND CANNOT BE MODIFIED
  - FUTURE MODIFICATIONS REQUIRE THE CREATION OF A NEW REVISION (CURRENT VERSION UNDER DEVELOPMENT)

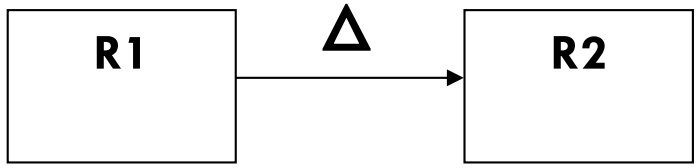
STORAGE OF REVISIONS

Version control tools do not store physically all versions, just one (the first or the last)

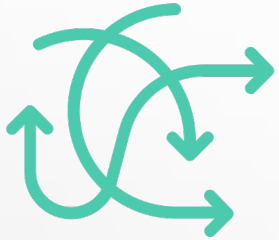
But any other version can be reconstructed from the change history

DELTA

- BEING R1 AND R2 TWO SUCCESSIVE REVISIONS IN THE EVOLUTION GRAPH
- A DELTA IS A SEQUENCE OF OPERATIONS THAT, WHEN APPLIED TO R1 GIVE AS A RESULT R2



TYPES OF DELTAS



According to Direction

DIRECT DELTAS take from a revision to the next one. Only the first revision is stored.

INVERSE DELTAS : take from a revision to the previous one. Only the last revision is stored (faster recovery of the current version)

According to Location

SEPARATED DELTAS: deltas are stored in separate files from the original document

MIXED DELTAS: they are stored within the same document

ALTERNATIVE CONFIGURATION

- Each Alternative Configuration is composed by a set of SCIs in a certain version
  - Definition
    - Association of attributes to each version of a SCI
    - Configuration specification describing the desired attributes
  - Building
    - Adequate scis and versions are recovered
    - The configuration is built

- MAKE, msbuild (visual studio), apache ant, apache maven (java): tools to facilitate building a specific configuration (recovery of scis, compiling and linking)

RELEASE

- A SYSTEM CONFIGURATION THAT IS COMMERCIALIZED OR DELIVERED TO THE CLIENT
- MUST BE IDENTIFIED AND STORED TO FACILITATE ITS FUTURE RECOVERY
- MANAGEMENT AND INSTALLING OF RELEASES IS A SCM ACTIVITY

SOFTWARE CONFIGURATION MANAGEMENT

- Basic concepts of SCM
- Configuration Identification
- Configuration Change Control
- Configuration Audit
- Configuration Status Accounting
- Configuration Management Plan

CONFIGURATION IDENTIFICATION

Goal

- Offer Visibility over the product
  - Components
  - Structure

Services

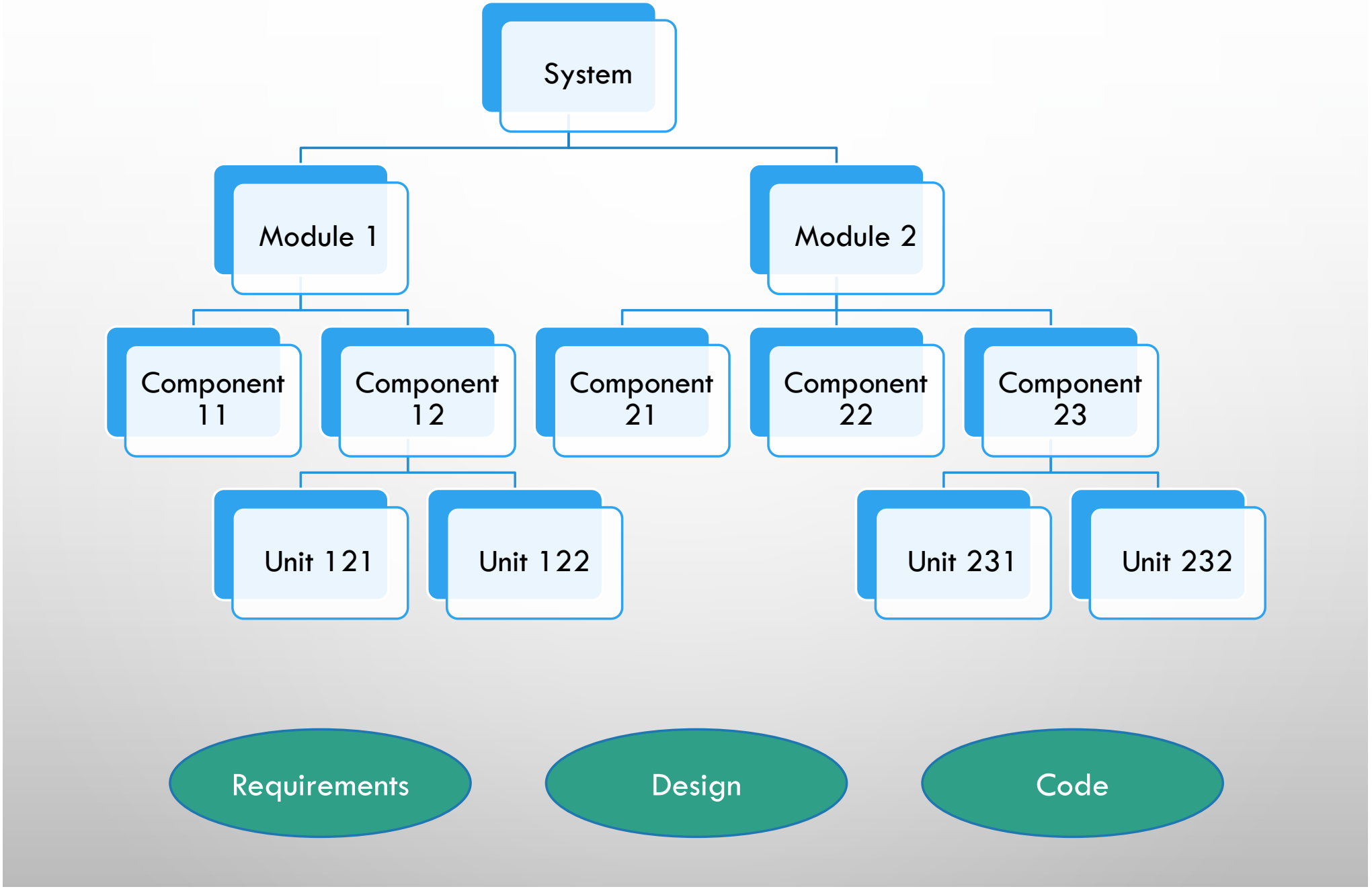
- Identification
- Accesibility

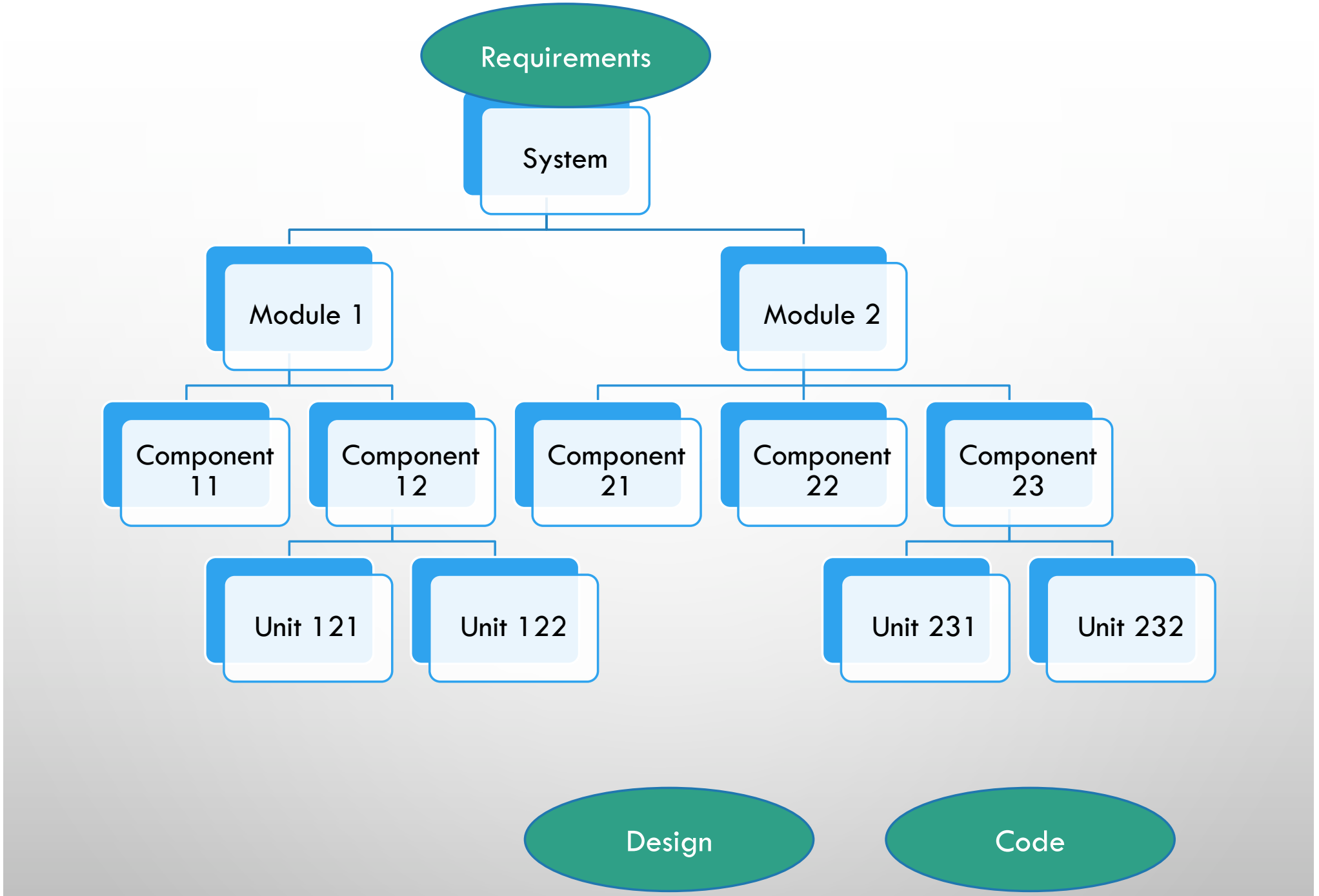
ACTIVITIES

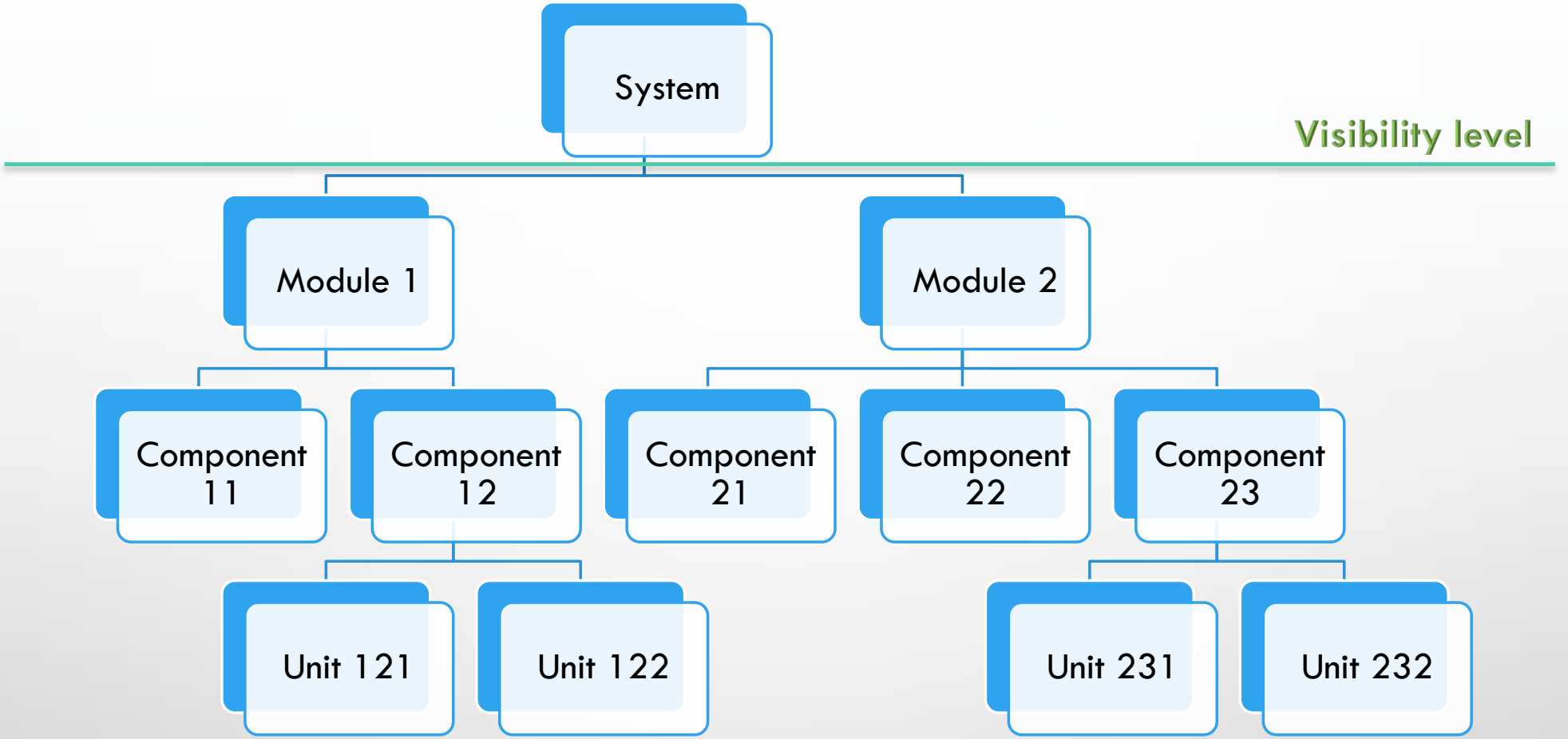
- PROJECT'S INITIATION
  - IDENTIFICATION
    - RELATED TO SCIS:
      - SELECTION OF SCIS
      - IDENTIFICATION OF THE COMPONENTS AND STRUCTURE OF THE SOFTWARE PRODUCT. VISIBILITY LEVEL
      - DEFINITION OF THE IDENTIFICATION SCHEMA
      - SELECTION OF RELATIONS IN THE CONFIGURATION TO MAINTAIN
    - RELATED TO BASELINES:
      - DEFINITION OF BASELINES TO ESTABLISH
  - ACCESIBILITY
    - DEFINITION OF SOFTWARE LIBRARIES TO USE

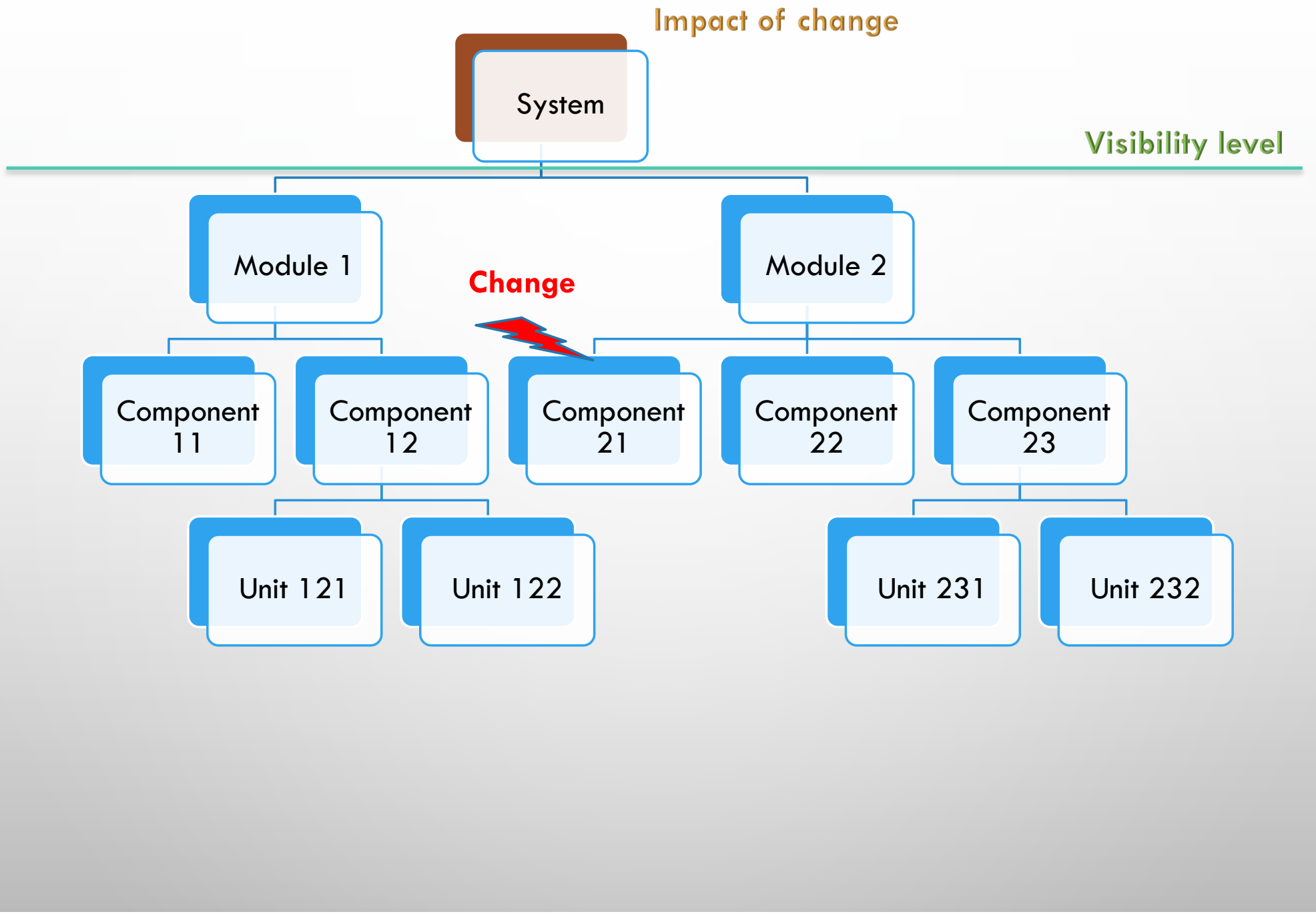
ACTIVITIES

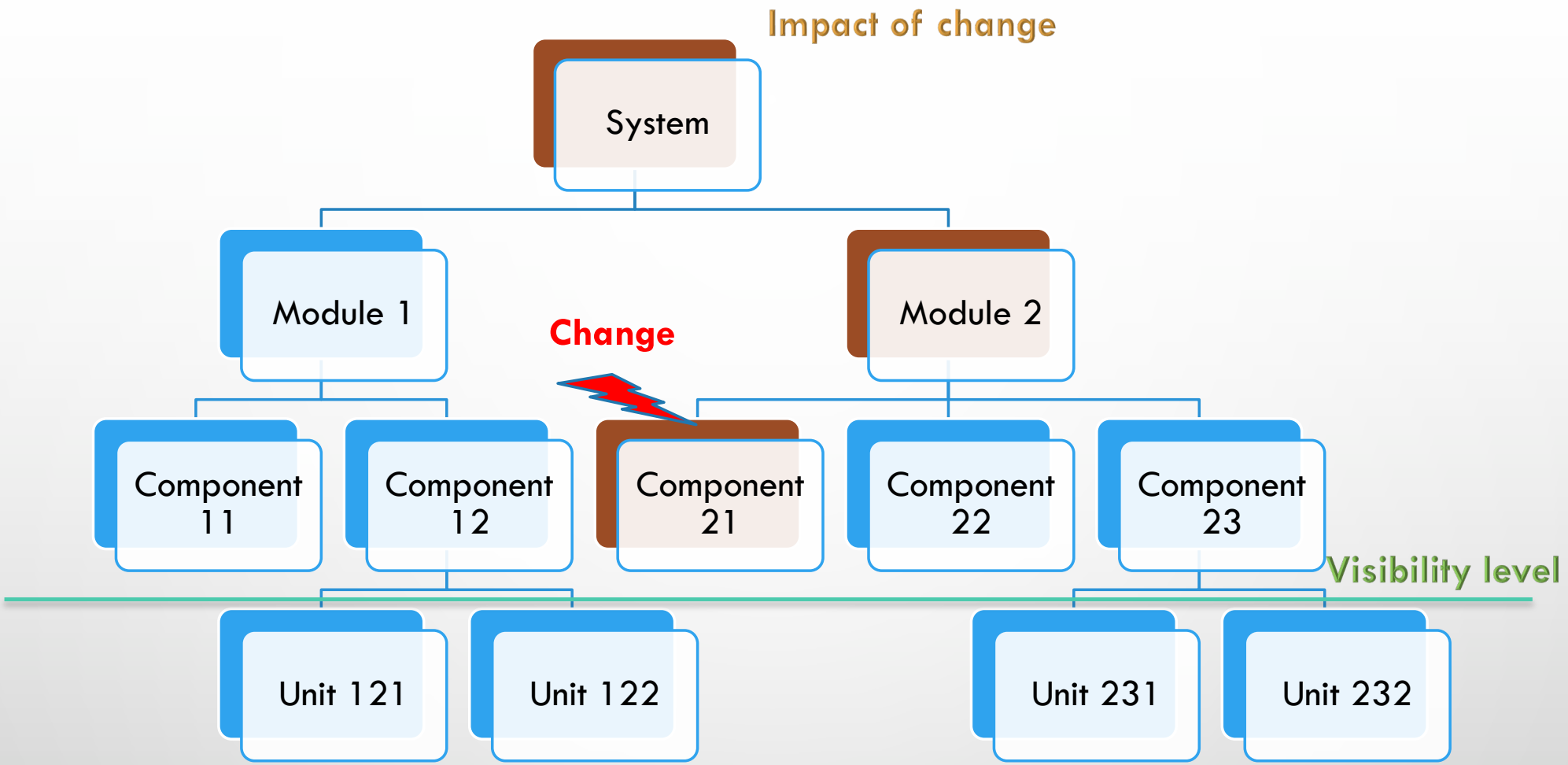
- PROJECT'S INITIATION
  - IDENTIFICATION
    - RELATED TO SCIS:
      - SELECTION OF SCIS
      - IDENTIFICATION OF THE COMPONENTS AND STRUCTURE OF THE SOFTWARE PRODUCT. VISIBILITY LEVEL
      - DEFINITION OF THE IDENTIFICATION SCHEMA
      - SELECTION OF RELATIONS IN THE CONFIGURATION TO MAINTAIN
    - RELATED TO BASELINES:
      - DEFINITION OF BASELINES TO ESTABLISH
  - ACCESIBILITY
    - DEFINITION OF SOFTWARE LIBRARIES TO USE


System
Module 1
Module 2
Component 11
Component 12
Component 21
Component 22
Component 23
Unit 121
Unit 122
Unit 231
Unit 232
Requirements
Design
Code


Requirements
System
Module 1
Module 2
Component 11
Component 12
Component 21
Component 22
Component 23
Unit 121
Unit 122
Unit 231
Unit 232
Design
Code


System
Visibility level
Module 1
Module 2
Component 11
Component 12
Component 21
Component 22
Component 23
Unit 121
Unit 122
Unit 231
Unit 232


Impact of change
System
Visibility level
Module 1
Module 2
Change
Component 11
Component 12
Component 21
Component 22
Component 23
Unit 121
Unit 122
Unit 231
Unit 232


Impact of change
System
Module 1
Module 2
Change
Component 11
Component 12
Component 21
Component 22
Component 23
Visibility level
Unit 121
Unit 122
Unit 231
Unit 232

ACTIVITIES

- PROJECT'S INITIATION
  - IDENTIFICATION
    - RELATED TO SCIS:
      - SELECTION OF SCIS
      - IDENTIFICATION OF THE COMPONENTS AND STRUCTURE OF THE SOFTWARE PRODUCT. VISIBILITY LEVEL
      - DEFINITION OF THE IDENTIFICATION SCHEMA
      - SELECTION OF RELATIONS IN THE CONFIGURATION TO MAINTAIN
    - RELATED TO BASELINES:
      - DEFINITION OF BASELINES TO ESTABLISH
  - ACCESIBILITY
    - DEFINITION OF SOFTWARE LIBRARIES TO USE

THINGS TO BE IDENTIFIED

- SCIs
- Versions
- Variants
- Alternative Configurations
- Releases

ACTIVITIES

- PROJECT'S INITIATION
  - IDENTIFICATION
    - RELATED TO SCIS:
      - SELECTION OF SCIS
      - IDENTIFICATION OF THE COMPONENTS AND STRUCTURE OF THE SOFTWARE PRODUCT. VISIBILITY LEVEL
      - DEFINITION OF THE IDENTIFICATION SCHEMA
      - SELECTION OF RELATIONS IN THE CONFIGURATION TO MAINTAIN
    - RELATED TO BASELINES:
      - DEFINITION OF BASELINES TO ESTABLISH
  - ACCESIBILITY
      - DEFINITION OF SOFTWARE LIBRARIES TO USE

DEFINITION OF CONFIGURATION RELATIONSHIPS

SCIs interconnected by relationships

Why?

To delimit the impact of a change

INTERESTING RELATIONSHIPS

- COMPOSITION (AMONG SCIS)
  - IN CODE
  - IN DOCUMENTATION
- DERIVATION (AMONG SCIS)
  - SOURCE CODE – OBJECT CODE
  - TEST CASE – EXECUTION TRACE
  - DESIGN OF MODULE N – SOURCE CODE OF MODULE
- ANY OTHER DEPENDENCY (AMONG SCIS)
  - DATA MODEL – DFDS

INTERESTING RELATIONSHIPS

- SUCCESION (AMONG SCI VERSIONS)
  - EXPLICIT
  - IMPLICIT (IN NUMBERING SCHEMA)
- EQUIVALENCE (AMONG COPIES OF A VERSION OF A SCI)
  - COPY IN DISC = COPY IN PAPER
  - OPTIONS
    - COPY INVENTORY IDENTIFYING LOCATION
    - CONTROLLED VS NON CONTROLLED COPIES

ACTIVITIES

- PROJECT'S INITIATION
  - IDENTIFICATION
    - RELATED TO SCIS:
      - SELECTION OF SCIS
      - IDENTIFICATION OF THE COMPONENTS AND STRUCTURE OF THE SOFTWARE PRODUCT. VISIBILITY LEVEL
      - DEFINITION OF THE IDENTIFICATION SCHEMA
      - SELECTION OF RELATIONS IN THE CONFIGURATION TO MAINTAIN
    - RELATED TO BASELINES:
      - DEFINITION OF BASELINES TO ESTABLISH
  - ACCESIBILITY
    - DEFINITION OF SOFTWARE LIBRARIES TO USE

DEFINITION OF BASELINES

- THE SCM PLAN SHOULD DEFINE:
  - THE BASELINES TO CREATE
  - HOW BASELINES ARE TO BE CREATED:
    - THE EVENT THAT CREATES THE BASELINE
    - THE ITEMS THAT ARE TO BE CONTROLLED IN THE BASELINE
    - THE PROCEDURES USED TO ESTABLISH AND CHANGE THE BASELINE
    - THE AUTHORITY REQUIRED TO APPROVE CHANGES TO THE APPROVED BASELINED ITEMS

COMMON BASELINES

Functional	System Requirements Analysis	Problem Definition Cost/time System Requirements
Function Allocation	Software Requirements Analysis	SRS for each software component
Preliminary Design	Arquitectural Design	Arquitecture Test Plan
Design	Detailed Design	Detailed Design Implementation Plan Test Design
Product	Testing	Programs Test Reports
Operation	Deployment	User Manual Installation Manual Operation Manual

ACTIVITIES

- PROJECT'S INITIATION
  - IDENTIFICATION
    - RELATED TO SCIS:
      - SELECTION OF SCIS
      - IDENTIFICATION OF THE COMPONENTS AND STRUCTURE OF THE SOFTWARE PRODUCT. VISIBILITY LEVEL
      - DEFINITION OF THE IDENTIFICATION SCHEMA
      - SELECTION OF RELATIONS IN THE CONFIGURATION TO MAINTAIN
    - RELATED TO BASELINES:
      - DEFINITION OF BASELINES TO ESTABLISH
  - ACCESIBILITY
    - DEFINITION OF SOFTWARE LIBRARIES TO USE

CONTROLLED SOFTWARE LIBRARIES

- CONTROLLED COLLECTION OF SOFTWARE AND/OR DOCUMENTATION
- TO BE DECIDED FOR EACH PROJECT
  - LIBRARIES TO USE
  - PEOPLE INVOLVED
  - PROCEDURES FOR
    - STORAGE
    - RETRIEVAL
  - TRANSITIONS AMONG LIBRARIES

COMMON LIBRARIES FOR A PROJECT

Working areas

- SCI under development
- Informal change

Integration area

- SCI components are assembled before delivery
- No changes

Project library

- Approved baselines
- Semi-formal change

Master library

- End of project and release
- Formal change

COMMON ORGANIZATIONAL LIBRARIES

SW Repository

- Retired products
- No change

Reusable components repository

- No change

Shared by all projects

ACTIVITIES

- DURING DEVELOPMENT
  - IDENTIFY / LABEL SCIS PRODUCED
  - ESTABLISH BASELINES
  - MAINTAIN RELATIONSHIPS IN CONFIGURATION
  - MAINTAIN SOFTWARE LIBRARIES
- NOT TAKING CHANGE INTO ACCOUNT YET

SOFTWARE CONFIGURATION MANAGEMENT

- Basic concepts of SCM
- Configuration Identification
- Configuration Change Control
- Configuration Audit
- Configuration Status Accounting
- Configuration Management Plan

TYPES OF CHANGES

Defect correction

System improvement

Configuration Change Control

LEVELS OF CONTROL

Informal

- Before the SCI enters a baseline
- No approval

Semiformal

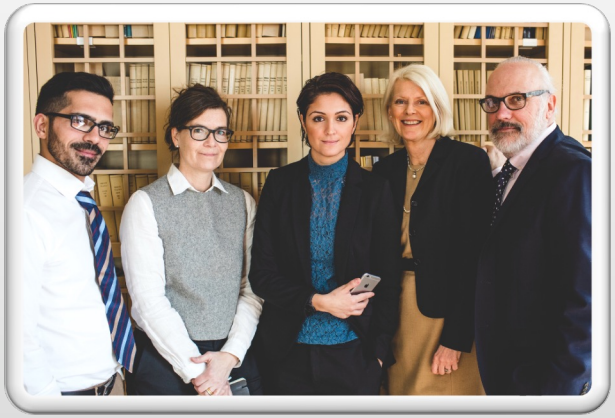
- After approval of SCI in baseline, before release
- Approval of Project Manager if local impact
- Approval of Change Control Board (CCB) if extended impact

Formal

- After transfer of SCIs to Master Library and release
- Approval by CCB

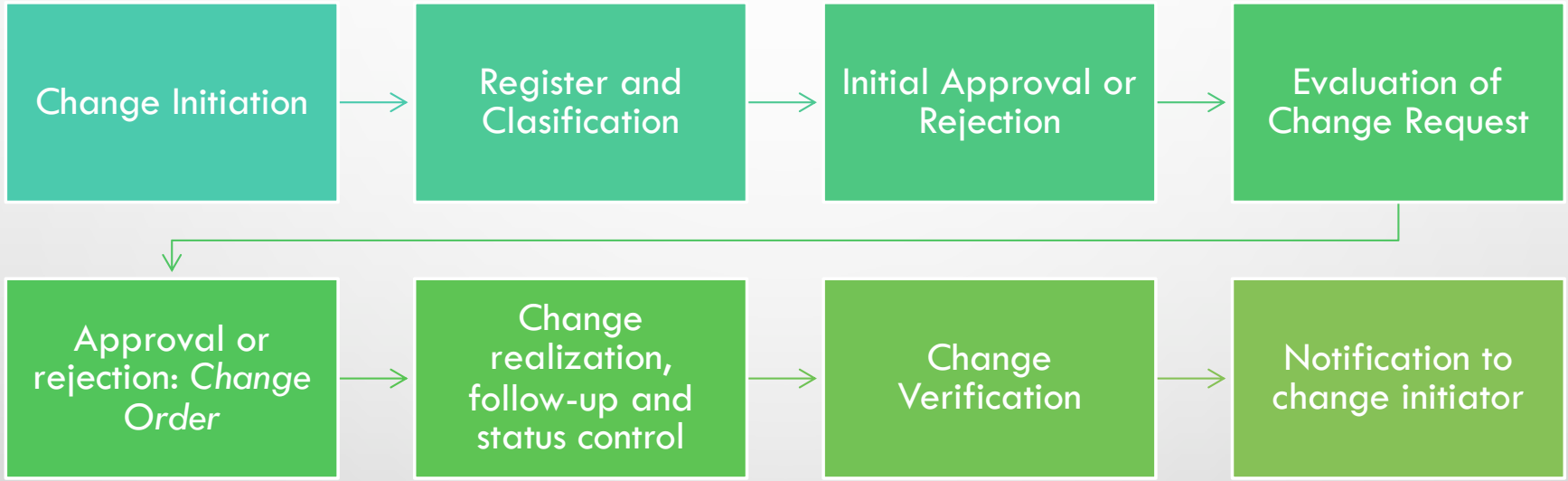
CHANGE CONTROL BOARD

- PERSON OR GROUP
- AUTHORITY ON **APPROVAL OR REJECTION** OF CHANGE REQUESTS
- REPRESENTATIVES FROM ALL STAKEHOLDERS
- NEED A GLOBAL VISION OF THE PRODUCT TO **EVALUATE THE IMPACT** OF A CHANGE
  - USUALLY DELEGATED TO A MORE TECHNICAL COMMITTEE
    - DESIGN REVIEW BOARD (DRB)
    - ENGINEERING REVIEW BOARD (ERB)
    - CHANGE ADVISORY BOARD (CAB)

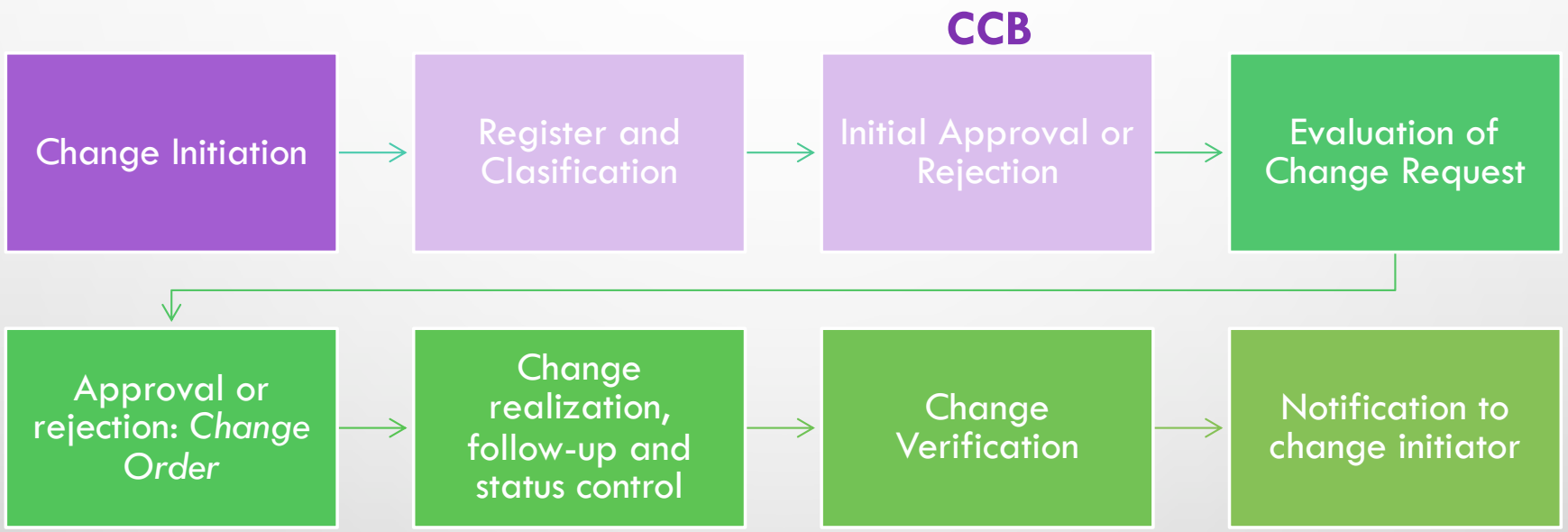


FORMAL CONTROL PROCESS

SEMI-FORMAL: Less bureaucracy



FORMAL CONTROL PROCESS



CHANGE INITIATION (PROBLEM)

Configuration Change Control

- (TEST) INCIDENT REPORT
  - DATE AND TIME
  - INFORMANT DATA
  - DESCRIPTION OF INCIDENT
  - EFFECTS
  - REPLICATION PROCEDURE
  - TEST CASE EXECUTED
  - MEMORY DUMP
  - ETC.

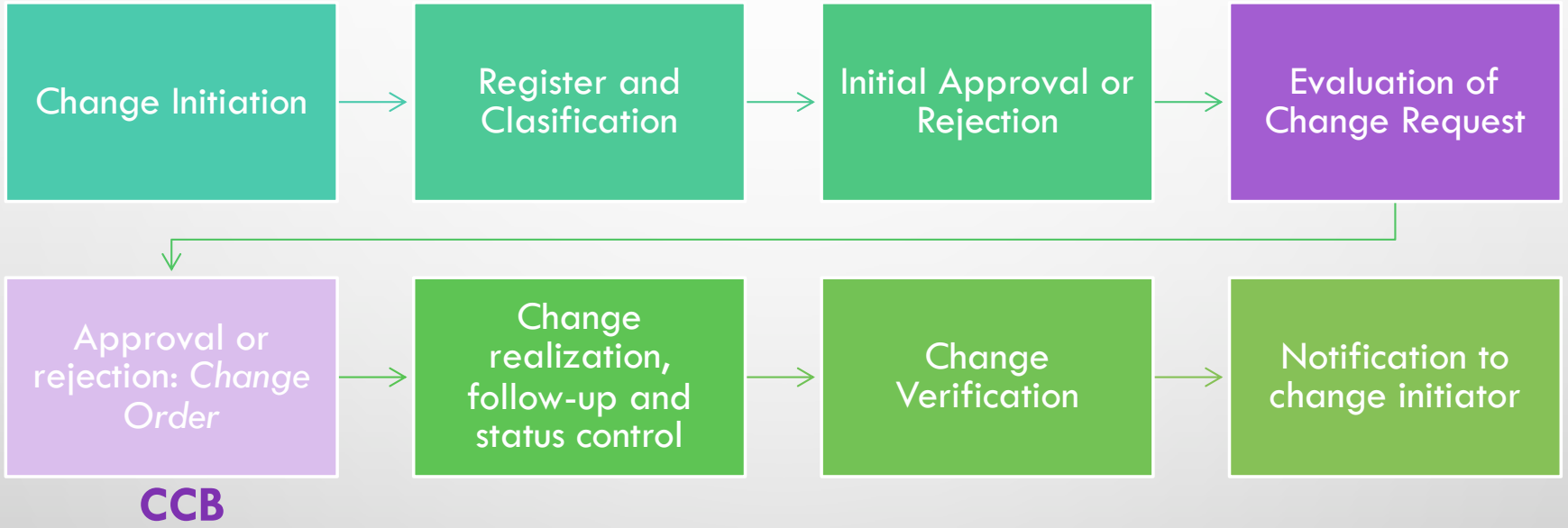
CHANGE INITIATION (IMPROVEMENT)

- CHANGE REQUEST FORM (CR)
  - DESCRIPTION OF PROPOSED CHANGE AND RATIONALE/PURPOSE
  - STATE OF THE CHANGE (E.G., OPEN, APPROVED/REJECTED, IMPLEMENTED, TESTED)
  - AFFECTED BASELINE
  - OUTCOME OF ANALYSIS OF IMPACT ON THE PROJECT
  - RESOLUTION
  - APPROVALS

  (MINIMUM REQUIRED BY IEEE STANDARD)

  - OTHER RELATED CHANGES
  - ALTERNATIVES
  - PRIORITY
  - ETC.

FORMAL CONTROL PROCESS



EVALUATION OF CHANGE REQUESTS

- Importance for project/organization
- ROI (return on investment)
- Needed resources (human and material)
- Size
- Complexity
- Estimated time
- Impact on product's quality
- Relationship to other changes
- Relevant organizational policies
- Alternatives

Configuration Change Control

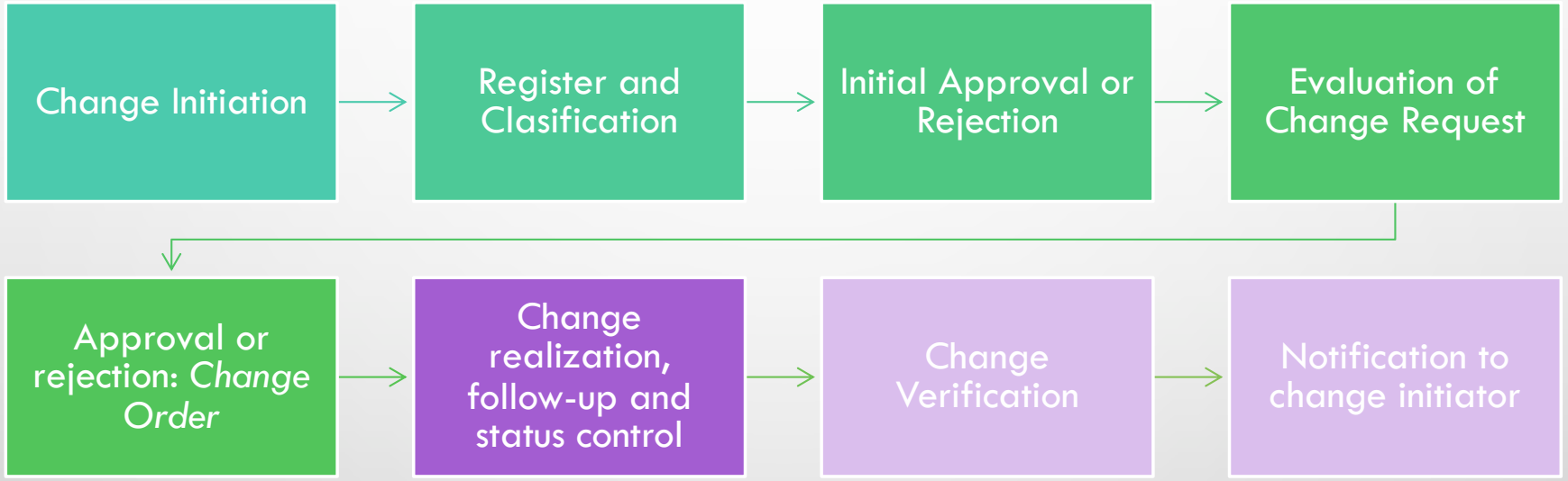
CHANGE ORDER

CHANGE TO BE PERFORMED

CONSTRAINTS

REVIEW AND AUDIT CRITERIA

FORMAL CONTROL PROCESS



CHANGE VERIFICATION

The defect was corrected, or the new requirements were addressed

No new defects appeared

Regresion Testing

COMPLETION OF A CHANGE

- CHANGE NOTIFICATION
  - THE ASSOCIATED CHANGE REQUEST(S) OR INCIDENT REPORT(S)
  - THE NAMES AND VERSIONS OF THE AFFECTED ITEMS
  - VERIFICATION DATE AND RESPONSIBLE PARTY
  - RELEASE OR INSTALLATION DATE AND RESPONSIBLE PARTY
  - THE IDENTIFIER OF THE NEW VERSION

(MINIMUM REQUIRED BY IEEE STANDARD)

  - DESCRIPTION OF CHANGE IMPLEMENTED
  - DATE
  - IMPLEMENTER

SOFTWARE CONFIGURATION MANAGEMENT

- Basic concepts of SCM
- Configuration Identification
- Configuration Change Control
- Configuration Audit
- Configuration Status Accounting
- Configuration Management Plan

RELATIONSHIP TO QUALITY CONTROL

Phase revisions	• Baseline Establishment
Change revisions	• Verify they were correctly performed
Audits	• Verify the integrity of the product (it is what it was meant to be)

Configuration Audit

SCM FUNCTIONS IN A PHASE REVIEW

- VERIFY THE CURRENT SOFTWARE CONFIGURATION WITH RESPECT TO THE PREVIOUS BASELINE (THE WORK WAS DONE WELL)
- VALIDATE THE CURRENT SOFTWARE CONFIGURATION WITH RESPECT TO REQUIREMENTS (INTEGRITY IS KEPT)
- ASSESS IF THE CURRENT SOFTWARE CONFIGURATION IS ACCEPTABLE OR NOT

PRODUCT AUDITS

SOFTWARE CONFIGURATION MANAGEMENT

- Basic concepts of SCM
- Configuration Identification
- Configuration Change Control
- Configuration Audit
- Configuration Status Accounting
- Configuration Management Plan

GOAL

Keep stakeholders informed of

ACTIVITIES

Information capture

- Where does it come from?
- What information to keep?

Storage

- Where is it stored?

Report generation

PRODUCTS

Records

Reports

To be decided for each project: *SCM Plan*

SOFTWARE CONFIGURATION MANAGEMENT

- Basic concepts of SCM
- Configuration Identification
- Configuration Change Control
- Configuration Audit
- Configuration Status Accounting
- Configuration Management Plan

SCM PLAN (IEEE 828-2005 STANDARD)

- INTRODUCTION
  - PROVIDES A SIMPLIFIED OVERVIEW OF THE SCM ACTIVITIES SO THAT THOSE APPROVING, THOSE PERFORMING, AND THOSE INTERACTING WITH SCM CAN OBTAIN A CLEAR UNDERSTANDING OF THE PLAN.
  - SHALL INCLUDE FOUR TOPICS:
    - THE PURPOSE OF THE PLAN: WHY THE PLAN EXISTS AND INTENDED AUDIENCE
    - THE SCOPE:
      - OVERVIEW DESCRIPTION OF THE SOFTWARE PROJECT
      - IDENTIFICATION OF THE SOFTWARE CI(S) TO WHICH SCM WILL BE APPLIED
      - IDENTIFICATION OF OTHER SOFTWARE TO BE INCLUDED AS PART OF THE PLAN (E.G., SUPPORT OR TEST SOFTWARE)
      - RELATIONSHIP OF SCM TO THE HARDWARE OR SYSTEM CONFIGURATION MANAGEMENT ACTIVITIES FOR THE PROJECT
      - THE DEGREE OF FORMALITY, DEPTH OF CONTROL, AND PORTION OF THE SOFTWARE LIFE CYCLE FOR APPLYING SCM ON THIS PROJECT
      - LIMITATIONS, SUCH AS TIME CONSTRAINTS, THAT APPLY TO THE PLAN
      - ASSUMPTIONS THAT MIGHT HAVE AN IMPACT ON THE COST, SCHEDULE, OR ABILITY TO PERFORM DEFINED SCM ACTIVITIES (E.G., ASSUMPTIONS OF THE DEGREE OF CUSTOMER PARTICIPATION IN SCM ACTIVITIES OR THE AVAILABILITY OF AUTOMATED AIDS).
    - THE DEFINITION OF KEY TERMS: A GLOSSARY OR REFERENCE TO A GLOSSARY
    - REFERENCES

SCM PLAN (IEEE 828-2005 STANDARD)

- SCM MANAGEMENT
  - DESCRIBES THE ALLOCATION OF RESPONSIBILITIES AND AUTHORITIES FOR SCM ACTIVITIES, AND THEIR MANAGEMENT, TO ORGANIZATIONS AND INDIVIDUALS WITHIN THE PROJECT STRUCTURE
  - SHALL INCLUDE FOUR TOPICS:
    - ORGANIZATION:
      - ALL ORGANIZATIONAL UNITS THAT PARTICIPATE IN OR ARE RESPONSIBLE FOR ANY SCM ACTIVITY ON THE PROJECT, AND FOR THE PROBLEM RESOLUTION PROCESS. ORGANIZATION CHARTS, SUPPLEMENTED BY STATEMENTS OF FUNCTION, ROLES, AND RELATIONSHIPS
    - SCM RESPONSIBILITIES:
      - ALLOCATION OF SCM ACTIVITIES TO ORGANIZATIONAL UNITS. MATRIX
    - APPLICABLE POLICIES, DIRECTIVES AND PROCEDURES
    - MANAGEMENT OF THE SCM:
      - ORGANIZATIONAL UNITS RESPONSIBLE FOR THE OVERALL SCM PROCESS, AND FOR INDEPENDENT SURVEILLANCE OF SCM ACTIVITIES TO ENSURE COMPLIANCE WITH THE SCM PLAN. COSTS. RISKS

SCM PLAN (IEEE 828-2005 STANDARD)

- SCM ACTIVITIES
  - CONFIGURATION IDENTIFICATION
    - STATE HOW EACH CI AND ITS VERSIONS ARE TO BE UNIQUELY NAMED (IDENTIFICATION SCHEMA)
    - IDENTIFY THE PROJECT CONFIGURATION ITEMS (CI) AND THEIR STRUCTURES AT EACH PROJECT CONTROL POINT (BASELINES)
    - DESCRIBE THE ACTIVITIES PERFORMED TO DEFINE, TRACK, STORE, AND RETRIEVE CIS (LIBRARIES)

SCM PLAN (IEEE 828-2005 STANDARD)

- SCM ACTIVITIES
  - CONFIGURATION CHANGE CONTROL
    - MECANISMS TO INITIATE CHANGES:
      - FORMS, TOOLS
      - PROCEDURES
    - MECANISMS TO EVALUATE CHANGES:
      - PROCEDURES
      - CRITERIA
    - MECANISMS TO DECIDE ON CHANGES:
      - PROCEDURES
      - AUTHORITY
    - MECANISMS TO VERIFY APPROVED CHANGES

SCM PLAN (IEEE 828-2005 STANDARD)

- SCM ACTIVITIES
  - CONFIGURATION CHANGE CONTROL
    - PROBLEM MANAGEMENT MECANISMS
    - VERSION CONTROL MECANISMS
  - STATUS ACCOUNTING
    - RECORDS TO BE MAINTAINED
    - REPORTS TO BE GENERATED
    - PROCEDURES FOR INFORMATION CAPTURE, STORAGE AND PROCESSING

SCM PLAN (IEEE 828-2005 STANDARD)

- SCM ACTIVITIES
  - CONFIGURATION AUDITS
    - GOALS
    - SCIS TO AUDIT
    - PARTICIPANTS
    - PROCEDURE TO CONDUCT IT
    - PROCEDURE TO REGISTER DEFFICIENCIES
    - CHECKLISTS AND QUESTIONNAIRES TO USE
    - SCI APPROVAL CRITERIA

SCM PLAN (IEEE 828-2005 STANDARD)

- SCM ACTIVITIES
  - SUBCONTRACTOR/VENDOR CONTROL
    - ITEMS DEVELOPED OUTSIDE THE PROJECT ENVIRONMENT
      - SOFTWARE DEVELOPED BY CONTRACT
      - SOFTWARE ACQUIRED IN ITS FINISHED FORM
    - DEFINE THE ACTIVITIES TO INCORPORATE THE EXTERNALLY DEVELOPED ITEMS INTO THE PROJECT
  - RELEASE MANAGEMENT AND DELIVERY
- SCHEDULES, RESOURCES, PLAN MAINTENANCE

SCM STANDARD (IEEE 828-2012)

- THE PREVIOUS 2005 EDITION (NOW OBSOLETE) DEFINED ONLY THE CONTENTS OF A SOFTWARE CONFIGURATION MANAGEMENT PLAN.
- THE NEW 2012 EDITION ALSO ADDRESSES:
  - WHAT CM ACTIVITIES ARE TO BE DONE
  - WHEN THEY ARE TO HAPPEN IN THE LIFE CYCLE
  - WHAT PLANNING AND RESOURCES ARE REQUIRED

SCM STANDARD (IEEE 828-2012)