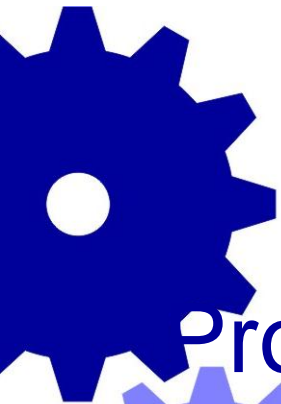## Rosana:

- Está em Lancaster desde agosto: trabalhando com Jaejoon
- Aluno de mestrado desistiu

Masiero

- Teste de programas OO/OA com ênfase em exceções e experimentos de custo e eficácia (Luciano). Possivel colaboração com Roberta e missão em novembro/dezembro
- Certificação de Processo de DS c/LPS
- CK+ Ger. Variabilidades em Simulink (Hephaestus e P:V) – Eduardo

Prolices: um processo de desenvolvimento de sistemas embarcados críticos

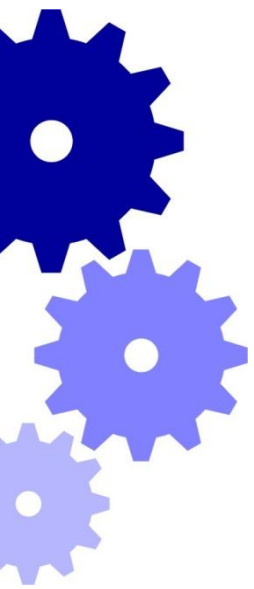## Rosana Braga
Masiero, Onofre, Kalinka...

Summary

- Introduction
- Background
- Previous Work
    - The Tiriba Product Line
    - SAFE-CRITES
- ProLiCES
- Preliminary results
- Conclusions

Summary

- Introduction
- Background
- Previous Work
  - The Tiriba Product Line
  - SAFE-CRITES
- ProLiCES
- Preliminary results
- Conclusions

Background: definitions

- Complex embedded system (at least four of the following features)
  - 1) multiprocessor or multi-core;
  - 2) 10k+ lines of code (without comment lines);
  - 3) multi-language;
  - 4) RTOS based;
  - 5) 10+ different types of I/O communication; and
  - 6) Critical nature.

Background: definitions

- UAS (Unmanned Aerial System) =
  aircraft + all associated elements (payload, ground control station, communication links)

- SPL (Software Product Line) = set of software systems that share common and managed features and fullls requirements of a particular market segment

- Model Driven Development (MDD): models are considered in all the development cycle and transformations are used to obtain lower level artifacts
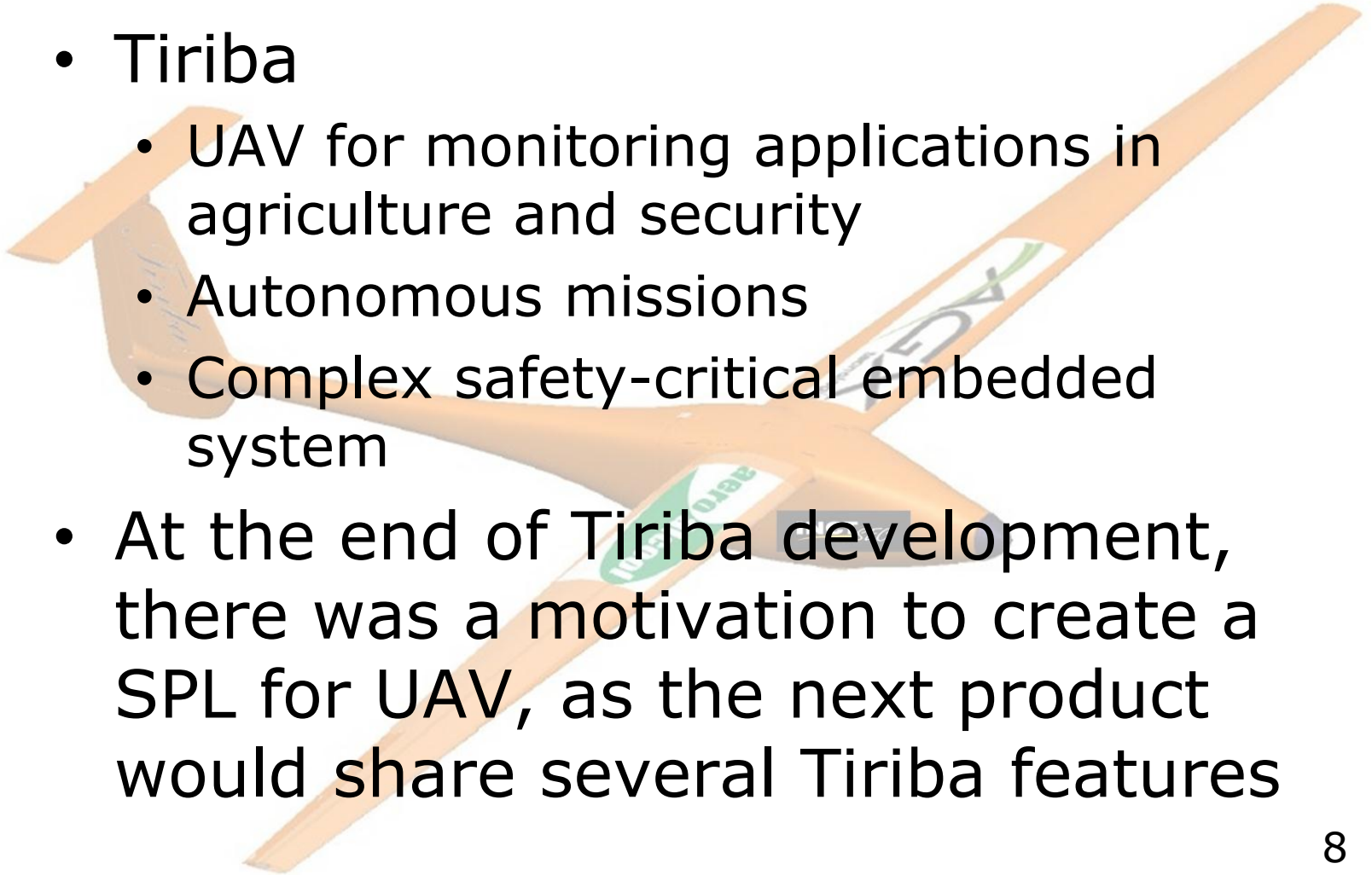
Summary

- Introduction
- Background
- Previous Work
  - The Tiriba Product Line
  - SAFE-CRITES
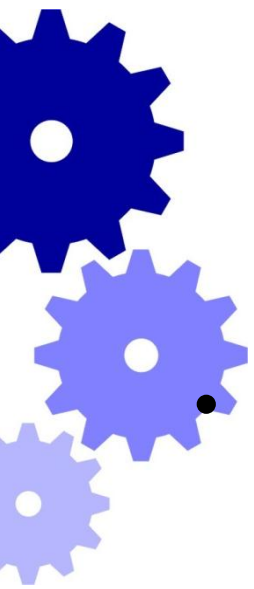- ProLiCES
- Preliminary results
- Conclusions

Previous work: Tiriba domain

- Tiriba
  - UAV for monitoring applications in agriculture and security
  - Autonomous missions
  - Complex safety-critical embedded system
- At the end of Tiriba development, there was a motivation to create a SPL for UAV, as the next product would share several Tiriba features

Previous work: SAFE-CRITES

- Process to develop embedded software based on reuse of artifacts
  - rigorous validation (extra effort to avoid the need of revising a previous phase)
  - Maximum use of modeling, simulation, and code generation tools
  - Maximum reuse of previously tested artifacts (not only code)
  - Not forcing the use of OO
  - Certification of the product (system) developed

SAFE-CRITES
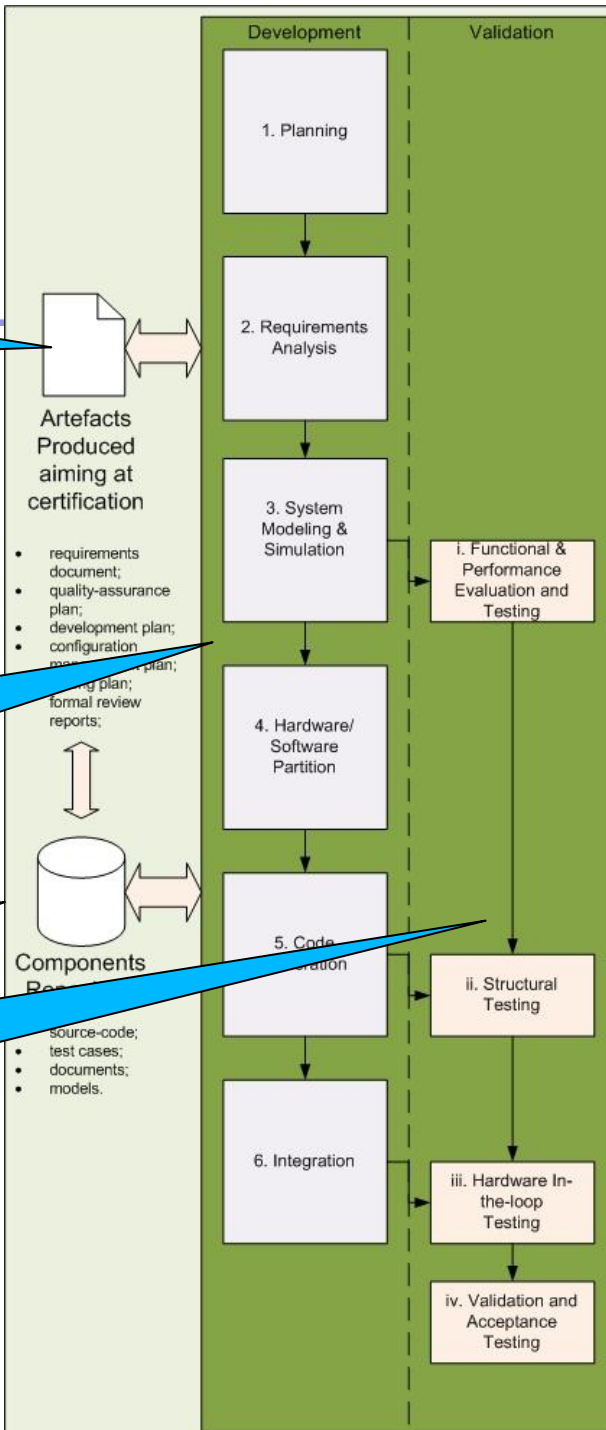
Artifacts produced are checked for certification

The process is composed of six development activities

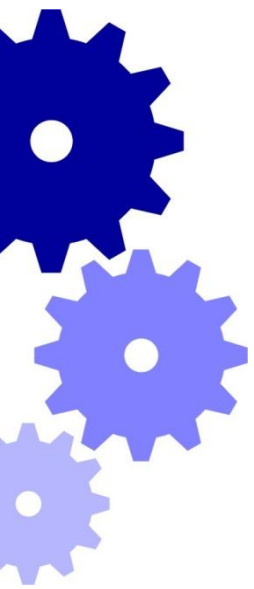A repository is created to ease reuse

Validation activities

Summary

- Introduction
- Background
- Previous Work
  - The Tiriba Product Line
  - SAFE-CRITES
- ProLiCES
- Preliminary results
- Conclusions

ProLiCES: motivation

- Tiriba development: potential software product line could be created
- Sarvant (new bigger project) has several features in common, but many other challenges

ProLiCES: overview

- SPL development:
  - domain engineering: reusable artifacts of the SPL are created
  - application engineering: artifacts are combined to produce concrete products
- Two-path parallel life cycle - which activity is done first depends on the context:
  - a concrete product can be created first, and then the SPL will be developed extractively or reactively based on one or more products; or
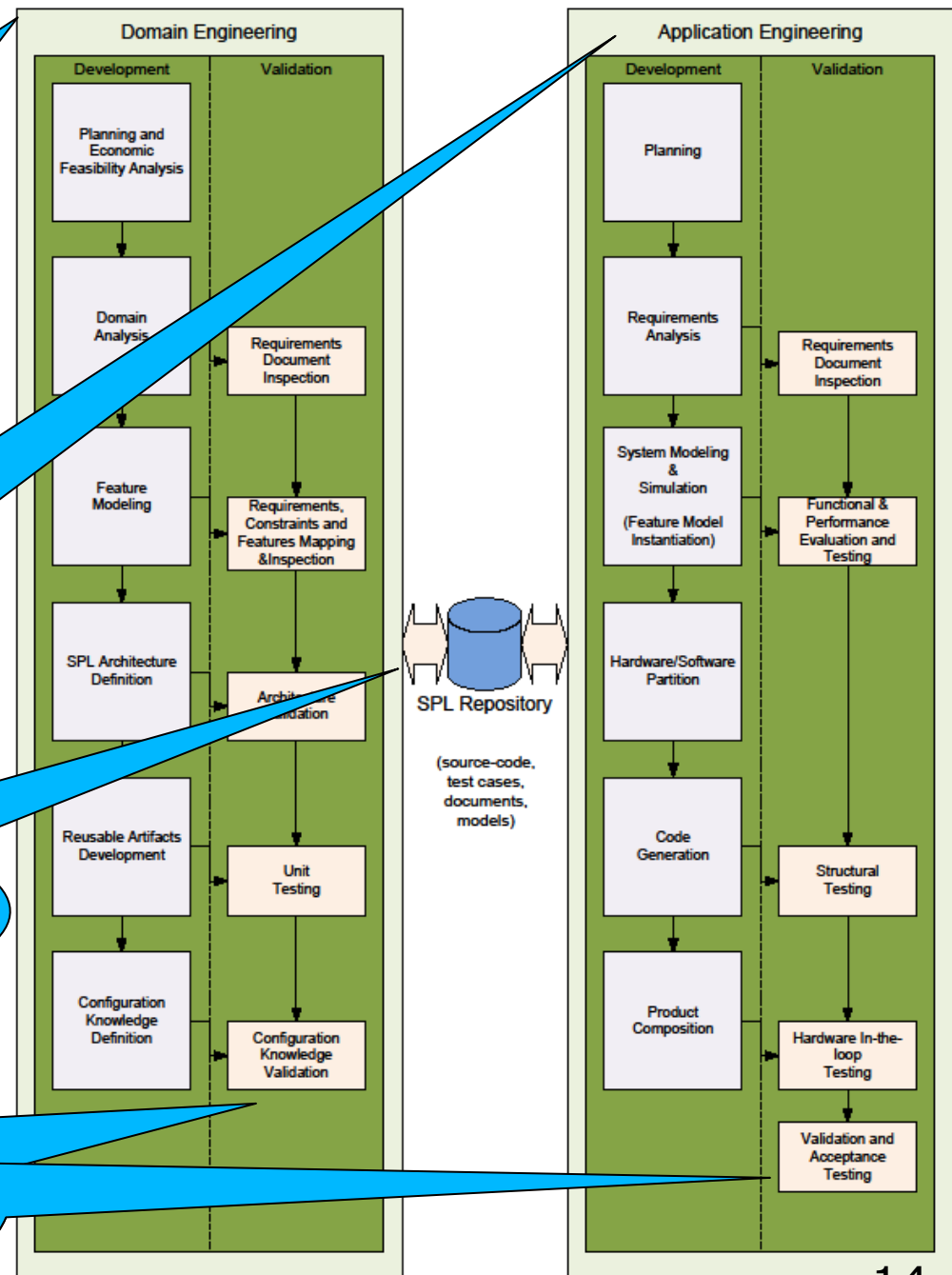 - the SPL can be developed in a proactive approach

ProLiCES

Domain Engineering

Application Engineering
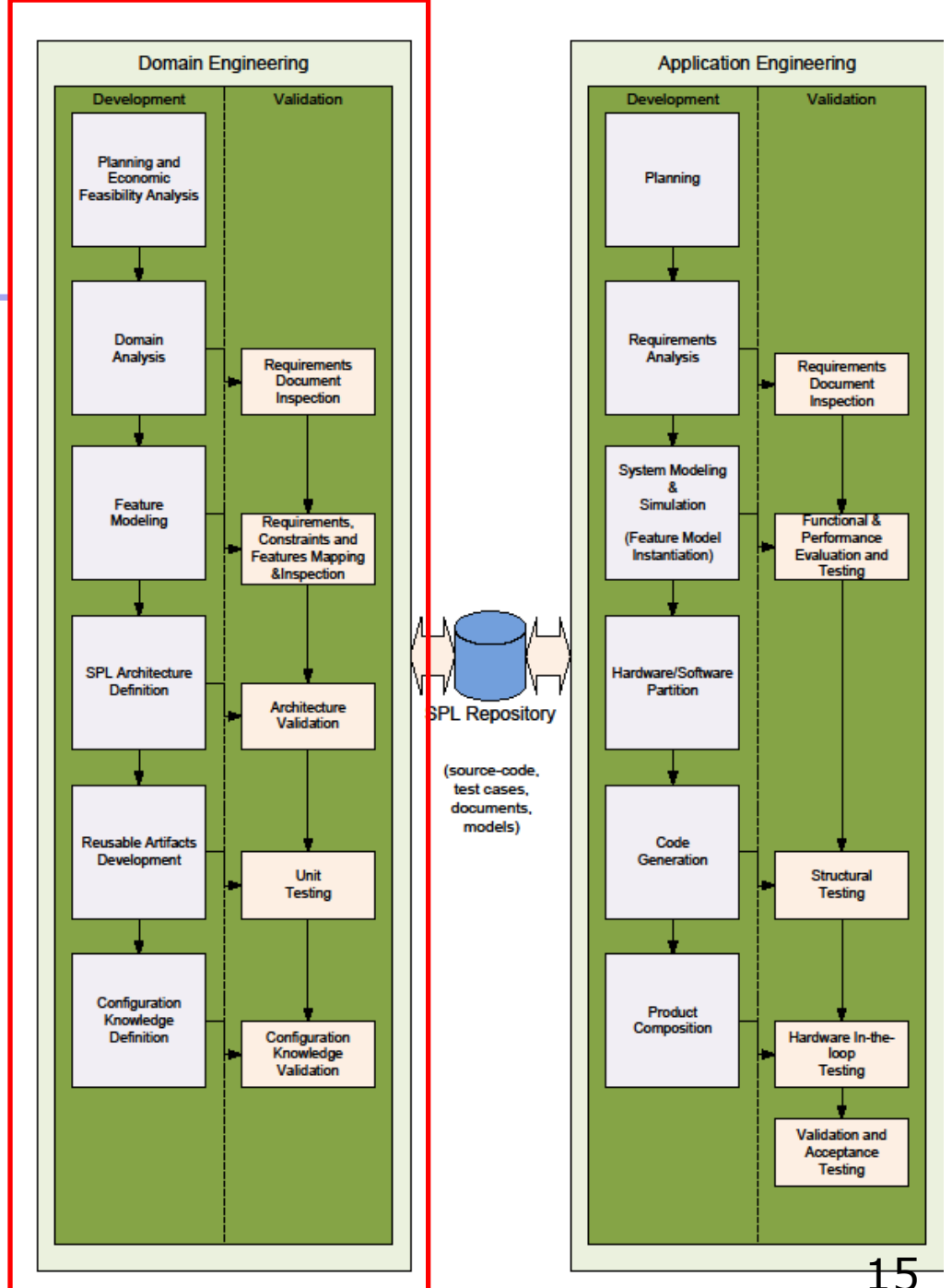
A repository of SPL is created to ease reuse

Verification & validation activities are present both in domain and in application engineering/

Domain Engineering

Development

- Planning and Economic Feasibility Analysis
- Domain Analysis
- Feature Modeling
- SPL Architecture Definition
- Reusable Artifacts Development
- Configuration Knowledge Definition

Validation

- Requirements Document Inspection
- Requirements, Constraints and Features Mapping &Inspection
- Architecture Validation
- Unit Testing
- Configuration Knowledge Validation

SPL Repository

(source-code, test cases, documents, models)

Application Engineering

Development

- Planning
- Requirements Analysis
- System Modeling & Simulation (Feature Model Instantiation)
- Hardware/Software Partition
- Code Generation
- Product Composition

Validation

- Requirements Document Inspection
- Functional & Performance Evaluation and Testing
- Structural Testing
- Hardware In-the-loop Testing
- Validation and Acceptance Testing

ProLiCES

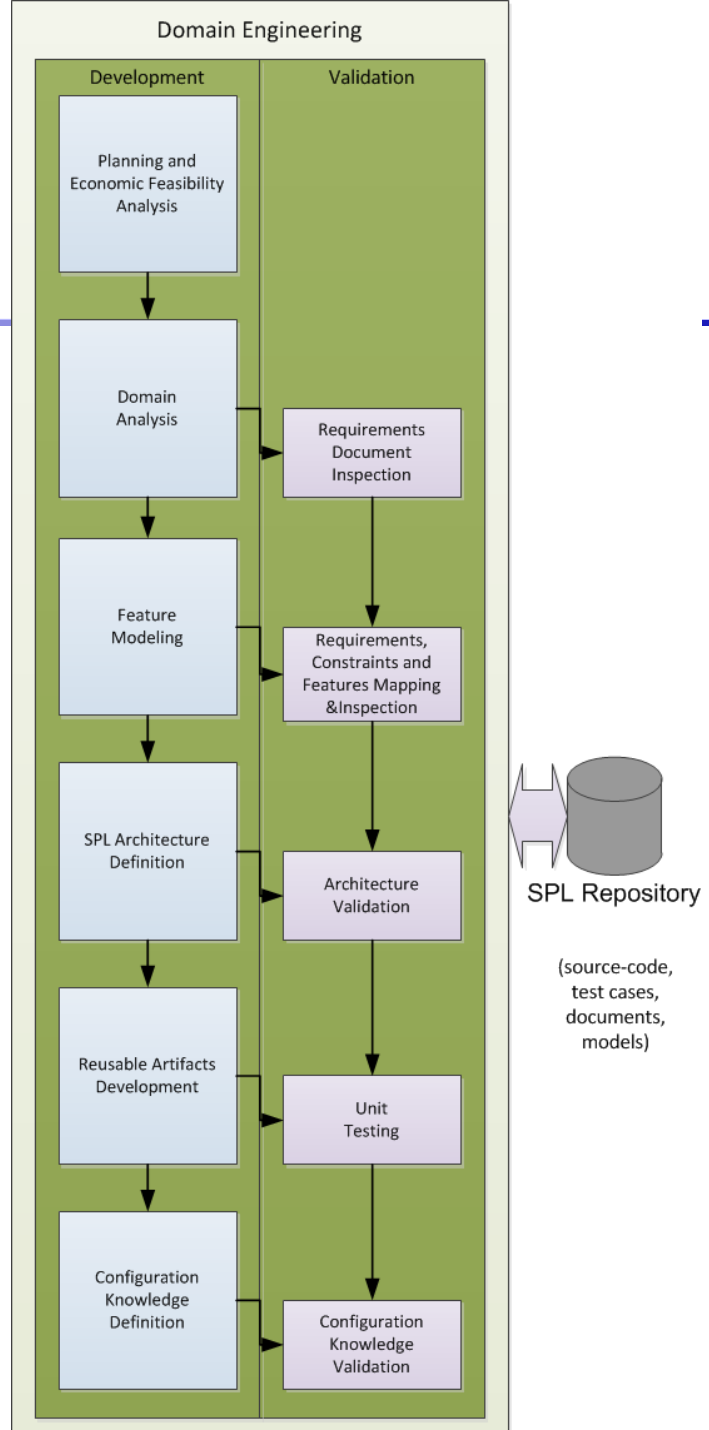## Domain Engineering

15

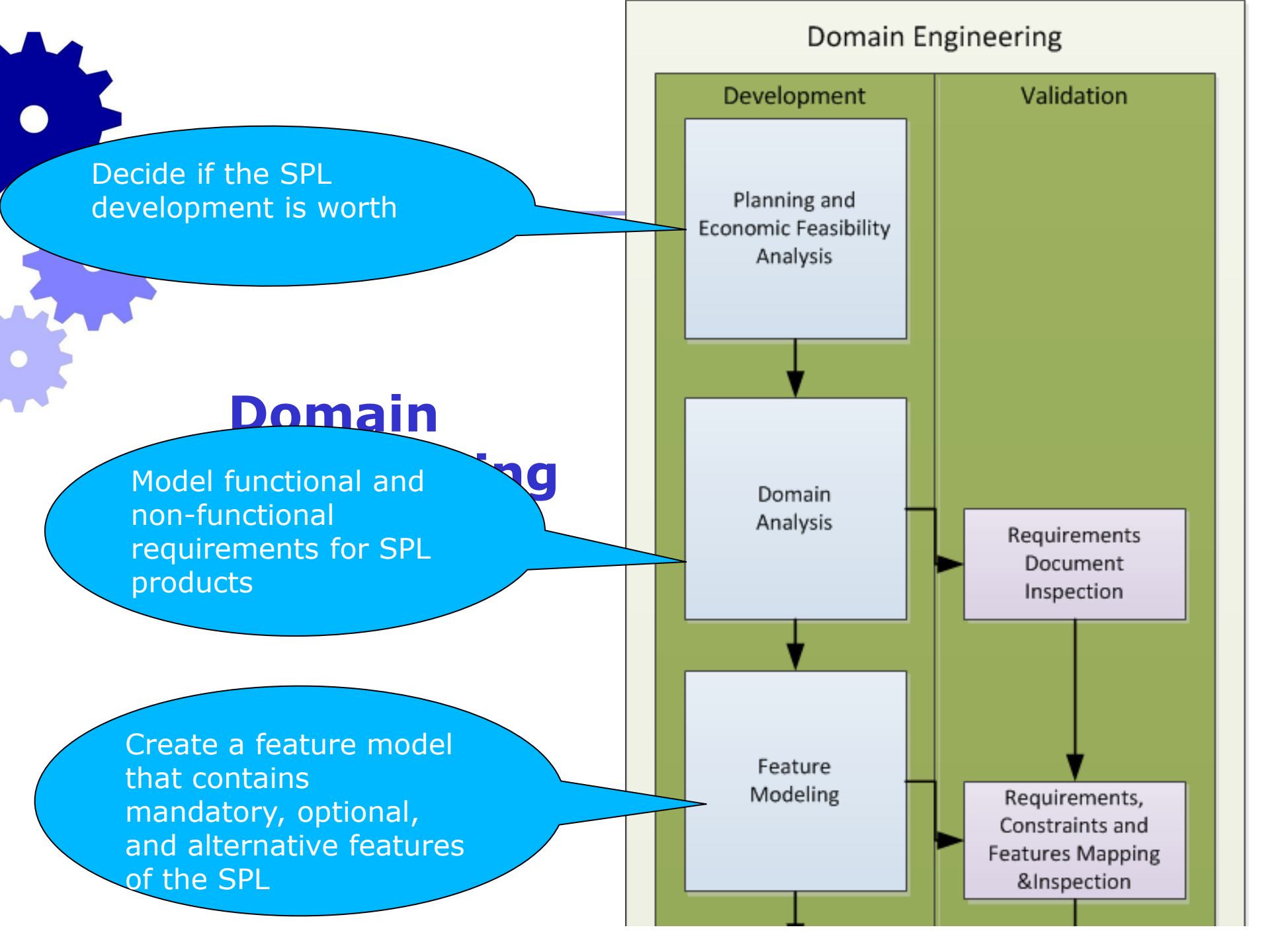# ProLiCES – Domain Engineering

- Main goal: produce reusable artifacts of the SPL
  - a set of core assets that will be present in any SPL product
  - a set of variable assets that will be included according to specific needs of the final product

Domain Engineering

Domain Engineering

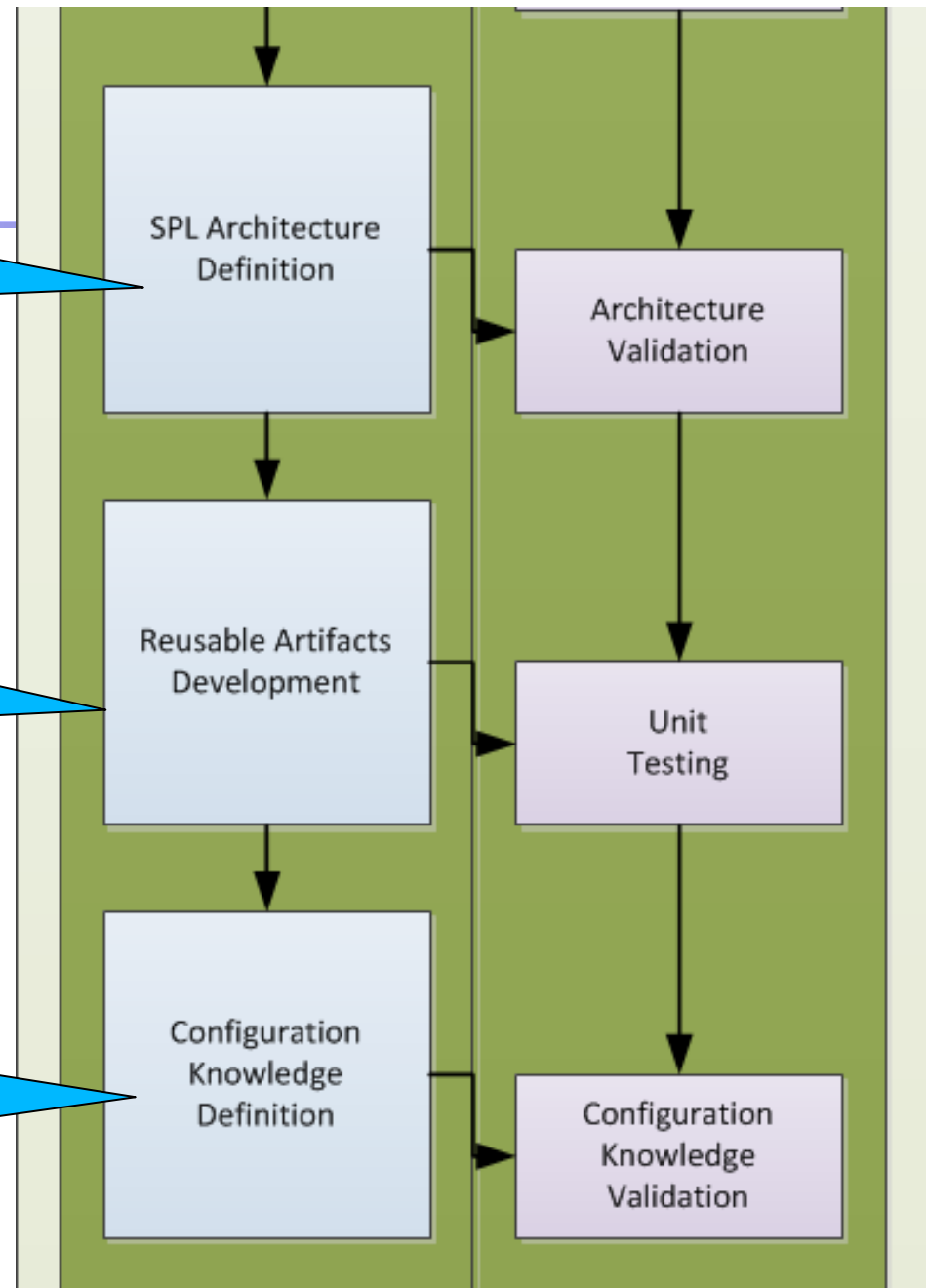| Development	Validation
Planning and Economic Feasibility Analysis	
Domain Analysis	Requirements Document Inspection
Feature Modeling	Requirements, Constraints and Features Mapping &Inspection
SPL Architecture Definition	Architecture Validation
Reusable Artifacts Development	Unit Testing
Configuration Knowledge Definition	Configuration Knowledge Validation

SPL Repository

(source-code, test cases, documents, models)

Domain Engineering

Decide if the SPL development is worth

Model functional and non-functional requirements for SPL products

Create a feature model that contains mandatory, optional, and alternative features of the SPL

Domain Engineering

Development

Planning and Economic Feasibility Analysis

Domain Analysis

Feature Modeling

Validation

Requirements Document Inspection

Requirements, Constraints and Features Mapping &Inspection

Domain Engineering

Identify the main components that will be built and the way how they relate to each other

SPL Architecture Definition

Architecture Validation

Develop reusable artifacts and store them at the repository

Reusable Artifacts Development

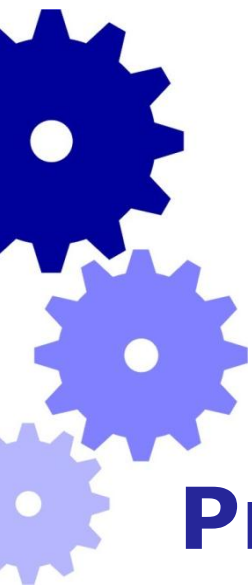Unit Testing

Map each feature of the SPL feature model to one or more artifacts that implement it.
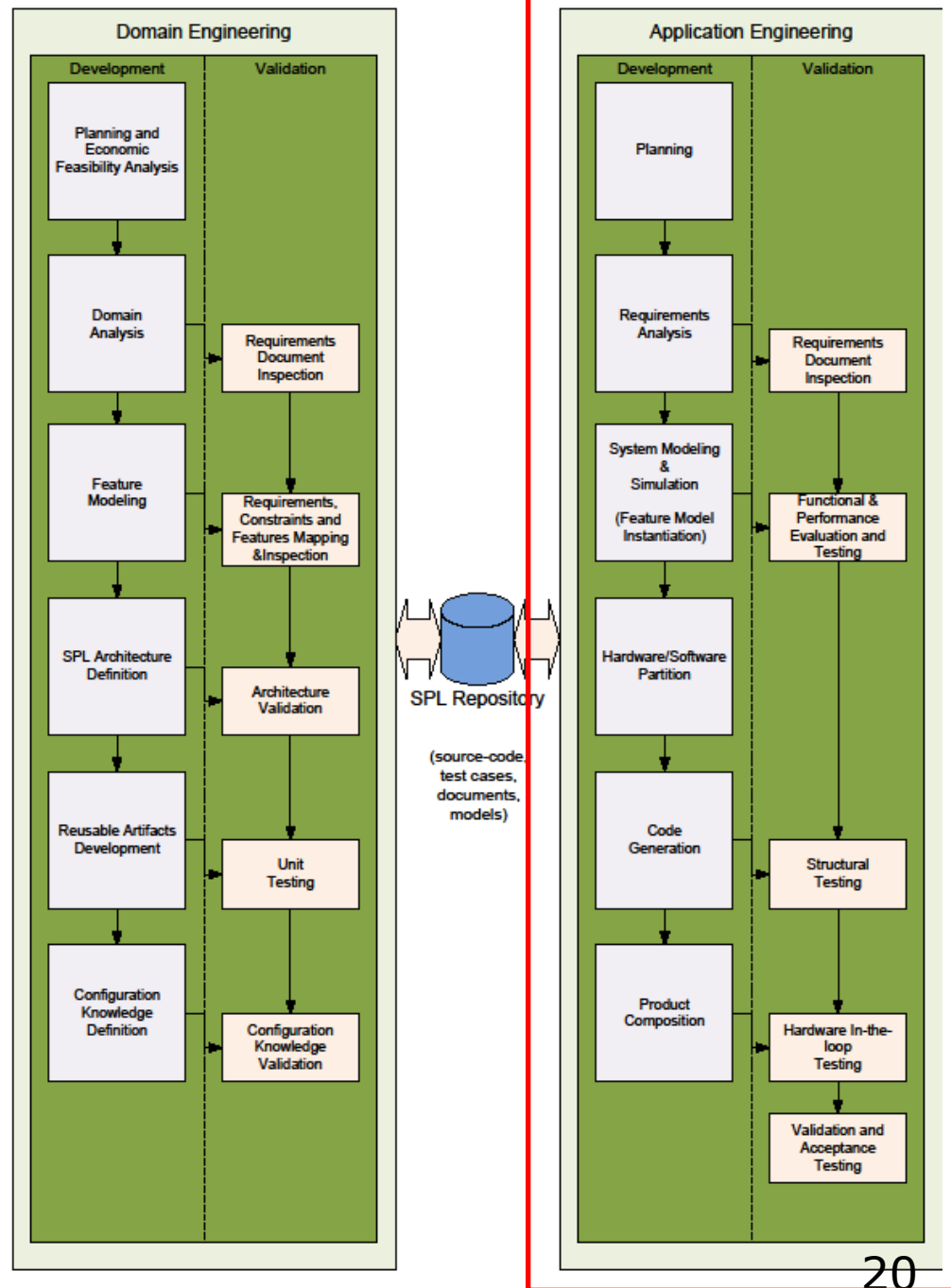
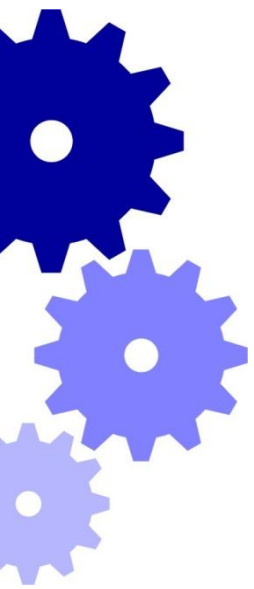Configuration Knowledge Definition

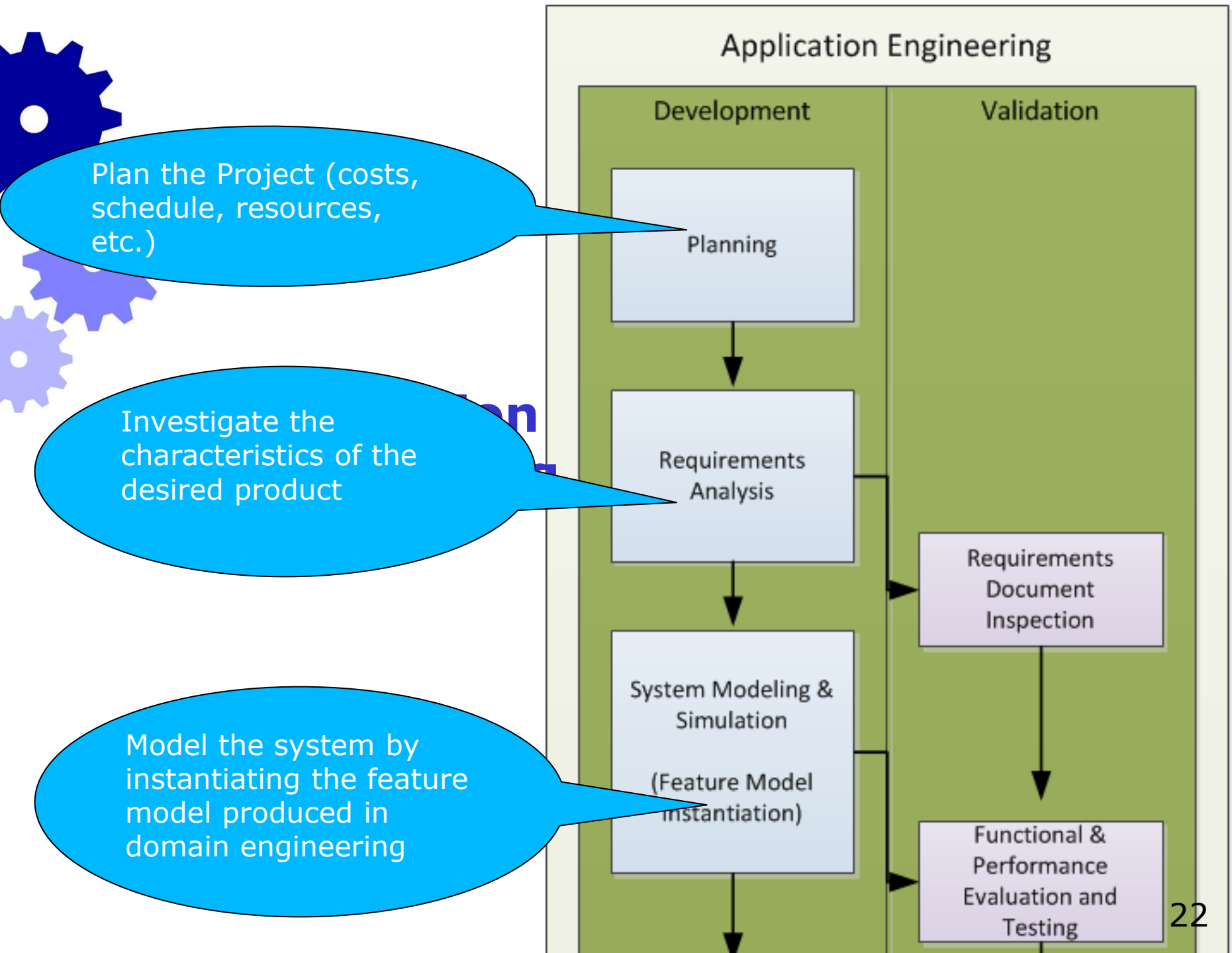Configuration Knowledge Validation
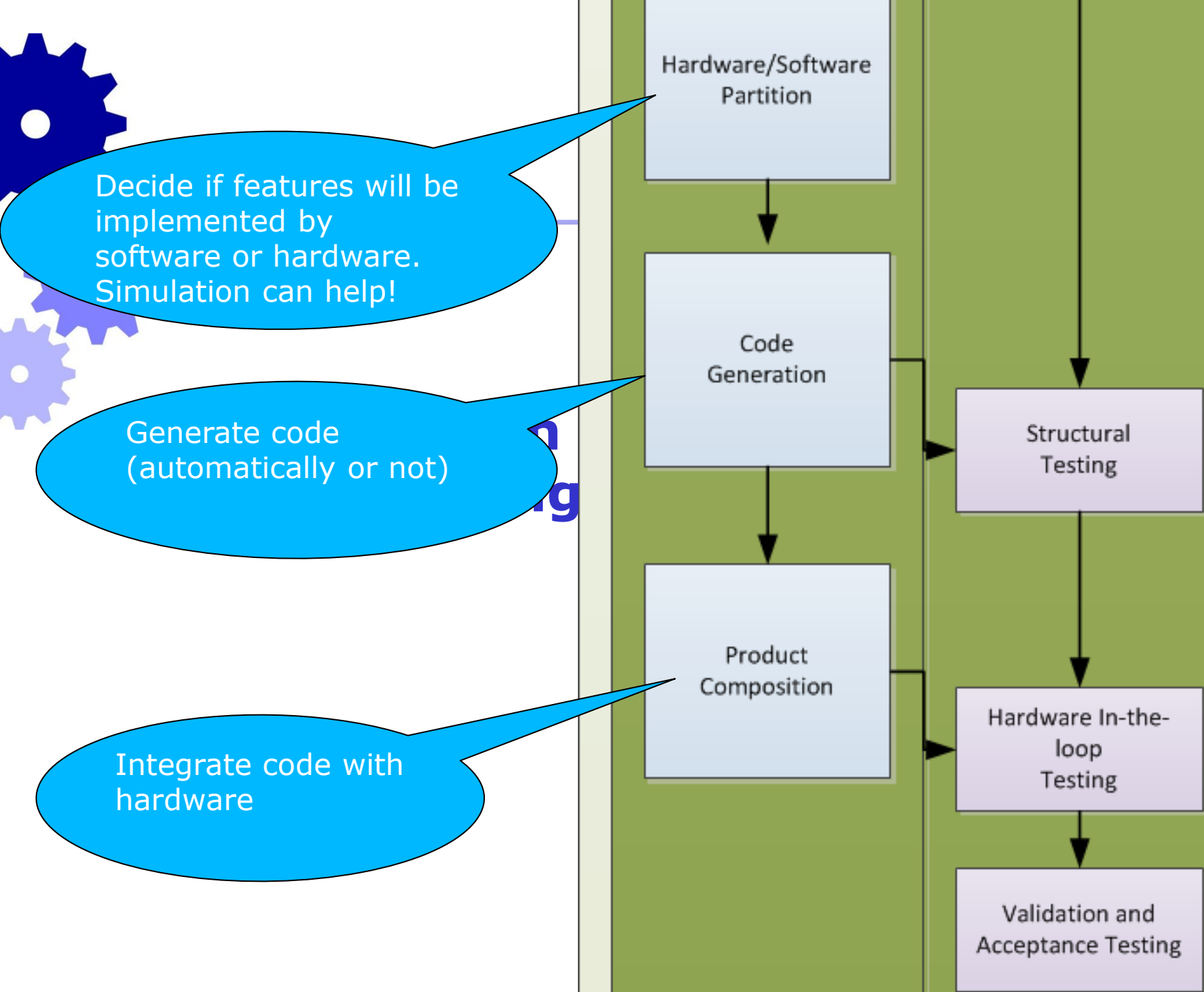
ProLiCES

Application Engineering

ProLiCES – Application Engineering

- Main goal: production of real-world SPL products.
  - Products are composed by selecting optional and alternative features

Application Engineering

Plan the Project (costs, schedule, resources, etc.)

Investigate the characteristics of the desired product

Model the system by instantiating the feature model produced in domain engineering

Development	Validation
Planning	
Requirements Analysis	Requirements Document Inspection
System Modeling & Simulation (Feature Model Instantiation)	Functional & Performance Evaluation and Testing

22

Hardware/Software Partition

Decide if features will be implemented by software or hardware. Simulation can help!

Code Generation

Generate code (automatically or not)

Structural Testing

Product Composition

Integrate code with hardware

Hardware In-the-loop Testing

Validation and Acceptance Testing

23

Summary

- Introduction
- Background
- Previous Work
    - The Tiriba Product Line
    - SAFE-CRITES
- ProLiCES
- Preliminary results
- Conclusions

Preliminary results

- ProLiCES is being used in the **Sarvant** project: development of a long-endurance unmanned aircraft with the specic mission of carrying a dual band SAR sensor

- Stakeholders:
  - AGX
  - OrbiSAT
  - INCT-SEC
  - FINEP

Preliminary results

- Tiriba was the basis for the feature model
- New features:
  - MOSA concept (Mission Oriented Sensor Array)
  - Redundancy of software and hardware
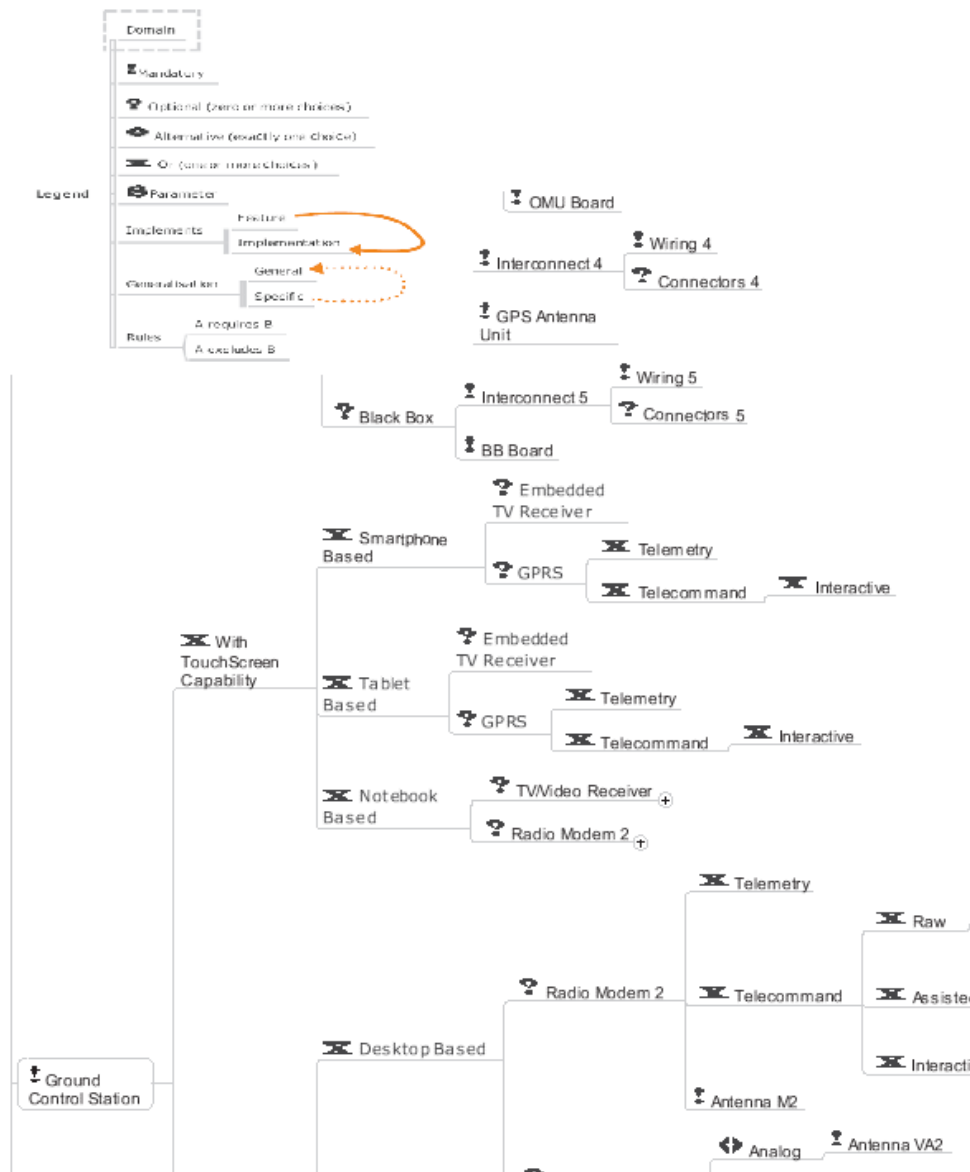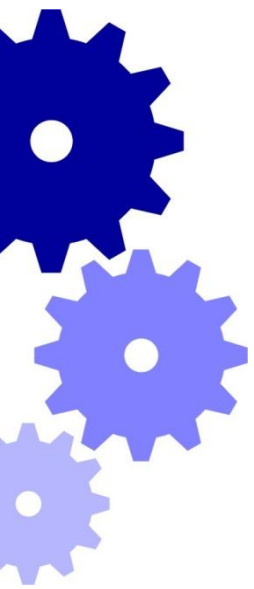 - IFA (In-Flight-Awareness)

Preliminary results

- Some results:
  - Feature model
  - System Architecture
  - Repository of reusable items
  - Hardware engineering

Preliminary results: Feature Model

Summary

○ Introduction

○ Background

○ Previous Work

      ○ The Tiriba Product Line

      ○ SAFE-CRITES

○ ProLiCES

○ Preliminary results

○ Conclusions

Conclusions

- ProLiCES applied to Tiriba project: 25.000 lines of code.

- Sarvant projetct: estimated 10 times bigger, two years to be concluded.

- Results of the application of ProLiCES to Sarvant will allow its refinement

- Certification under DO-178B is also a goal

Thanks!!

○ Contact by E-mail
  ● rtvb@icmc.usp.br (Rosana)