Assurance Based Development of Critical Systems

AUTHORS: P. GRAYDON, J. KNIGHT, E. STRUNK

PRESENTED BY: MIKE MAKSIMOV

Overview

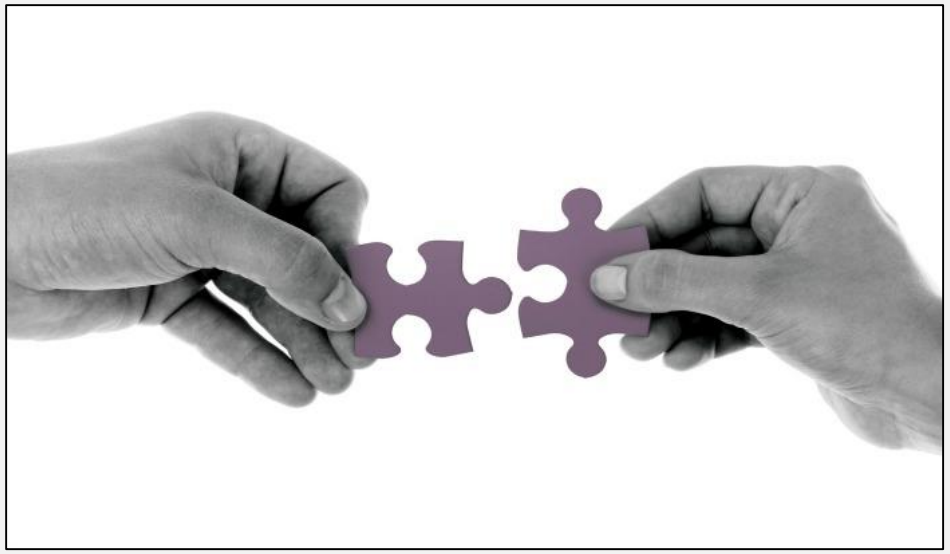
1. Introduction to Assurance Cases
2. Overview of the Problem
3. Assurance Based Development (ABD)
   - Candidate Development Choices
   - Selection of a System Development Choice
   - Applying System Development Choice
4. Illustrative Example of the ABD Process
5. Discussion

Introduction

- Definition of ABD – "*synergistic construction of a critical computing system and an assurance case….*"

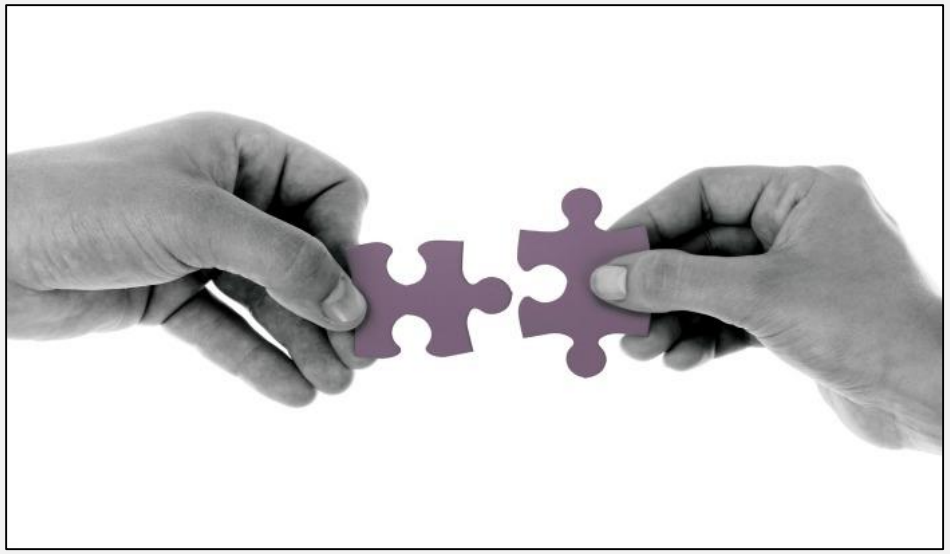
Introduction

- Definition of ABD – "*synergistic construction of a critical computing system and an assurance case….*"



Introduction

- Definition of ABD – "*synergistic construction of a critical computing system and an assurance case….*"
- Definition of an Assurance case – "a documented body of evidence that provides a convincing and valid argument that a specified set of critical claims regarding a system's properties are adequately justified for a given application in a given environment" Scott and Krombolz (2005)



Safety Case

- Safety Cases are a subset of Assurance Cases that argue the safety of a system.

Q: What do they look like?

Safety Case

- Safety Cases are a subset of Assurance Cases that argue the safety of a system.

Q: What do they look like?

A: It depends..

- We have various types:
  - Textual
  - Graphical

Within the **context** of the tolerability targets for hazards (from reference Z) and the list of hazards identified from the functional hazard analysis (from reference Y), we follow the **strategy** of arguing over all three of the identified hazards (H1, H2, and H3) to establish sub-claim 1, yielding three additional **claims**: H1 has been eliminated; H2 has been sufficiently mitigated; and H3 has been sufficiently mitigated.

The **evidence** that H1 has been eliminated is formal verification.

The **evidence** that catastrophic hazard H2 has been sufficiently mitigated is a fault tree analysis showing that its probability of occurrence is less than $1x10^{-6}$ per annum. The **justification** for using this evidence is that the acceptable probability in our environment for a catastrophic hazard is $1x10^{-6}$ per annum.

The **evidence** that the major hazard H3 has been sufficiently mitigated is a fault tree analysis showing that its probability of occurrence is less than $1x10^{-3}$ per annum. The **justification** for using this evidence is that the acceptable probability in our environment for a major hazard is $1x10^{-3}$ per annum.

We establish sub-claim (2) within the **context** of the list of hazards identified from the functional hazard analysis in reference Y, and the integrity level (IL) process guidelines defined in reference X. The process **evidence** shows that the primary protection system was developed to the required IL 4. The process **evidence** also shows that the secondary protection system was developed to the required IL 2.

Claim 1: Control system is acceptably safe.
Context 1: Definition of acceptably safe.

Claim 1.1: All identified hazards have been eliminated or sufficiently mitigated.
Context 1.1-a: Tolerability targets for hazards (reference Z).
Context 1.1-b: Hazards identified from functional hazard analysis (reference Y).

Strategy 1.1: Argument over all identified hazards (H1, H2, H3)

Claim 1.1.1: H1 has been eliminated.
Evidence 1.1.1: Formal verification

Claim 1.1.2: Probability of H2 occurring < $1x10^{-6}$ per annum.
Justification 1.1.2: $1x10^{-6}$ per annum limit for catastrophic hazards.
Evidence 1.1.2.: Fault Tree analysis.

Claim 1.1.3: Probability of H3 occurring < $1x10^{-3}$ per annum.
Justification 1.1.3: $1x10^{-3}$ per annum limit for major hazards.
Evidence 1.1.3: Fault tree analysis.

Claim 1.2: The software has been developed to the integrity level appropriate to the hazards involved.
Context 1.2-a: (same as Context 1.1-b)
Context 1.2-b: Integrity level (IL) process guidelines defined by reference X.

Claim 1.2.1: Primary protection system developed to IL 4.
Evidence 1.2.1: Process evidence of IL 4

Claim 1.2.2: Secondary protection system developed to IL 2.
Evidence 1.2.2: Process evidence of IL 2.

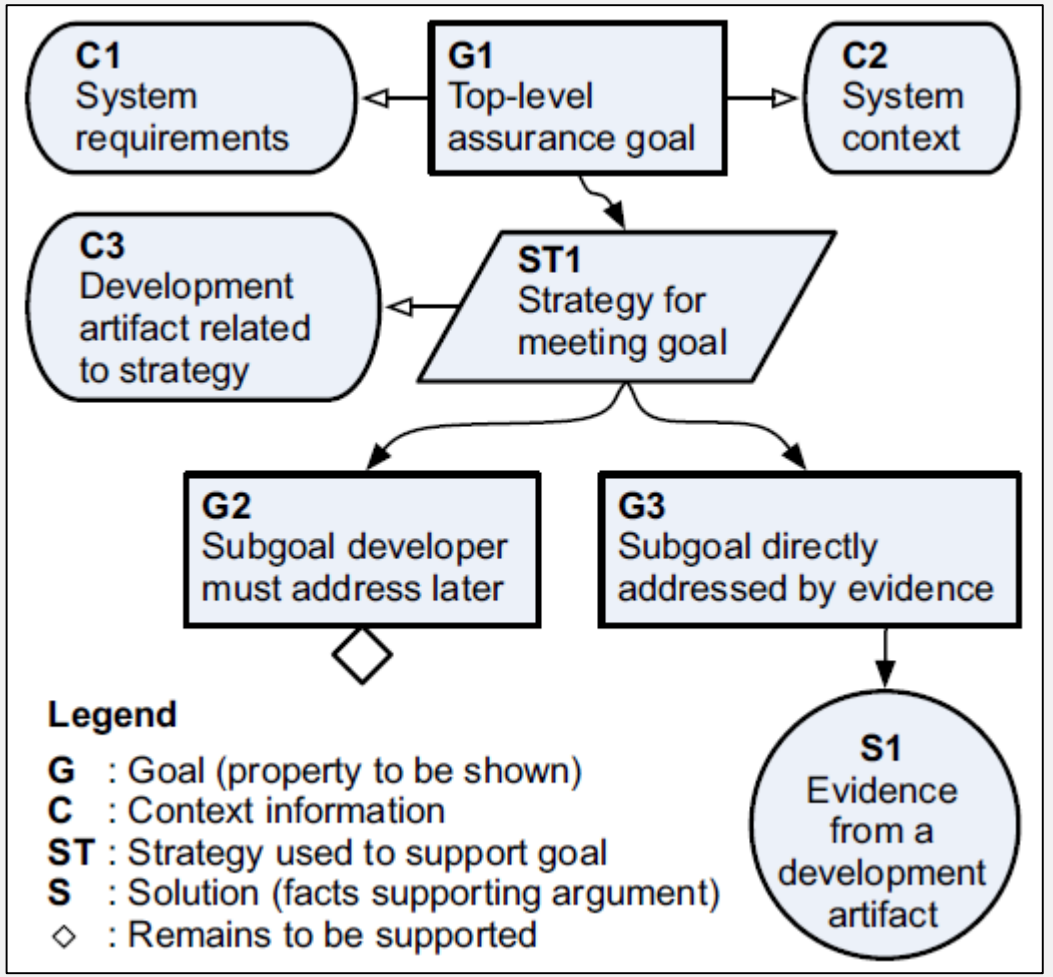
Safety Case

- Safety Cases are a subset of Assurance Cases that argue the safety of a system.

Q: What do they look like?

A: It depends..

- We have various types:
  - Textual
  - Graphical (Ex. GSN Notation)



Current Development Practices

- Current dependability assurance approaches are ad hoc.
- Developers carry out dependability testing on isolated units without being able to evaluate the ensuing effects to the system as a whole.
- Assurance cases produced at the end of development might not have enough evidence from the development process.

Current Development Practices

- ~~Current dependability assurance approaches are ad hoc.~~
- ~~Developers carry out dependability testing on isolated units without being able to evaluate the ensuing effects to the system as a whole.~~
- ~~Assurance cases produced at the end of development might not have enough evidence from the development process.~~

All of this can lead to the revisiting of development steps after the development process is complete!

Assurance Based Development

- Confidence that the system will meet its dependability goals is evaluated throughout the development process.

- The system and it’s assurance argument are co-developed so that the impacts of a development choice are available at the time the choice is made.

Assurance Based Development

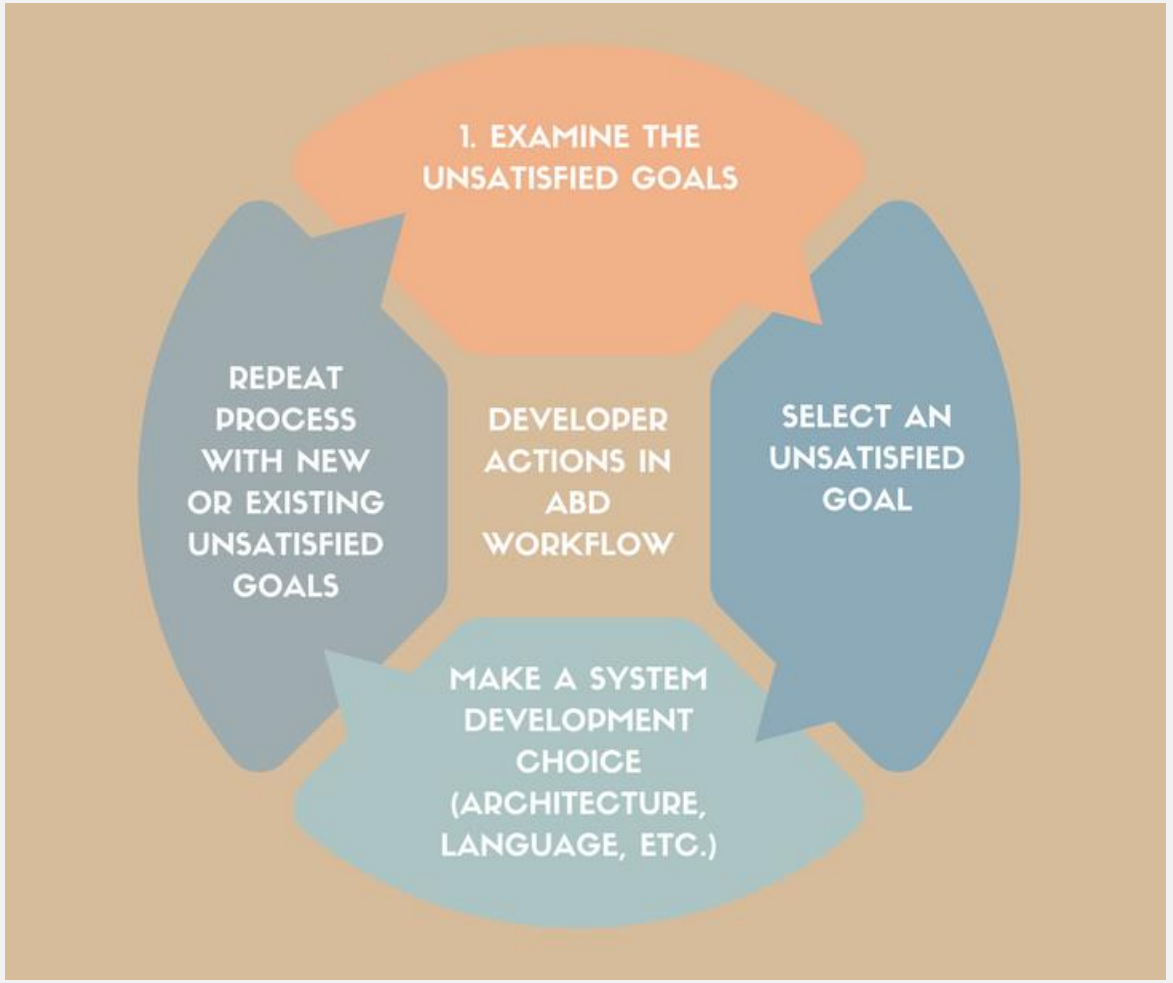
- Confidence that the system will meet its dependability goals is evaluated throughout the development process.
- The system and it's assurance argument are co-developed so that the impacts of a development choice are available at the time the choice is made.

- This helps avoid and detect potential assurance difficulties *as they arise*.
- The Assurance Case can be exploited to drive development choices.
- You have confidence that you have enough evidence to support your claims.
- You have confidence that you are producing a dependable product.

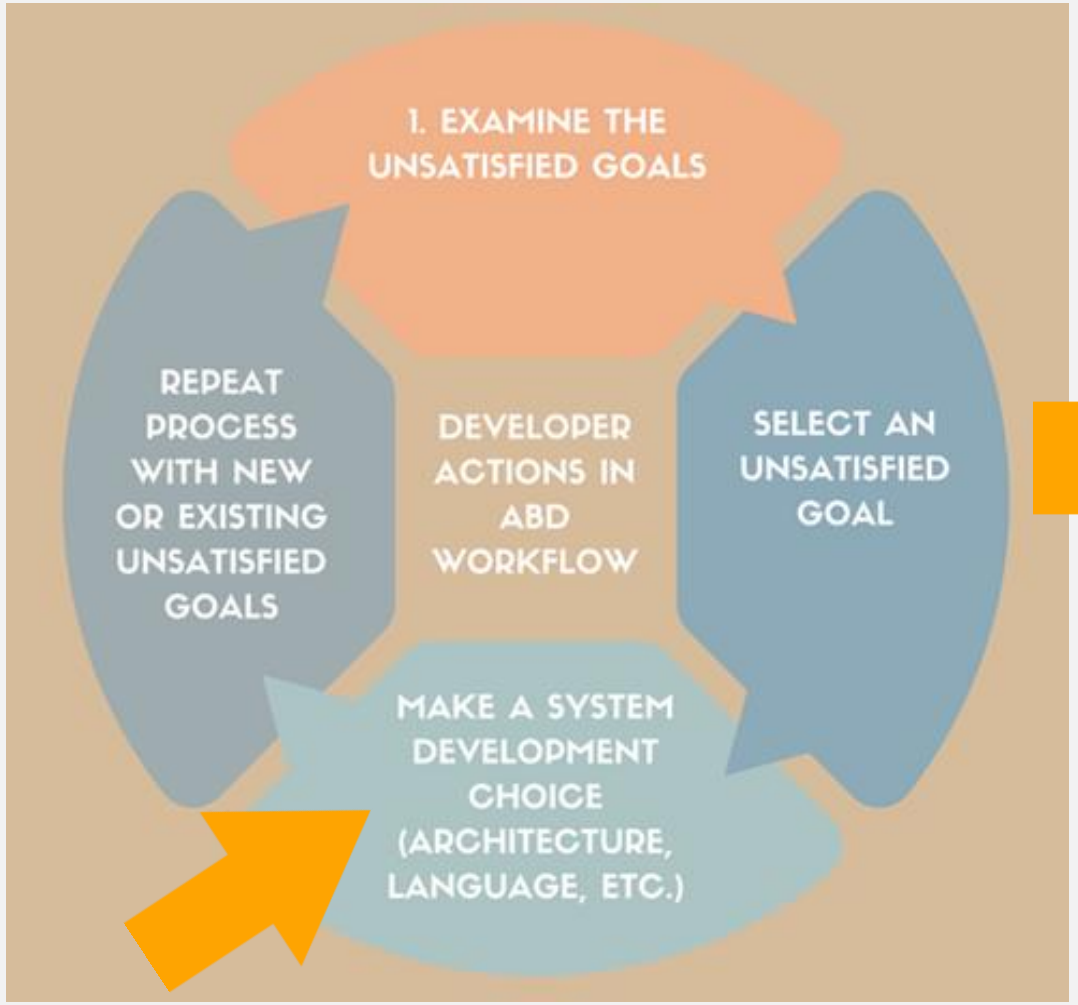
ABD Workflow Overview

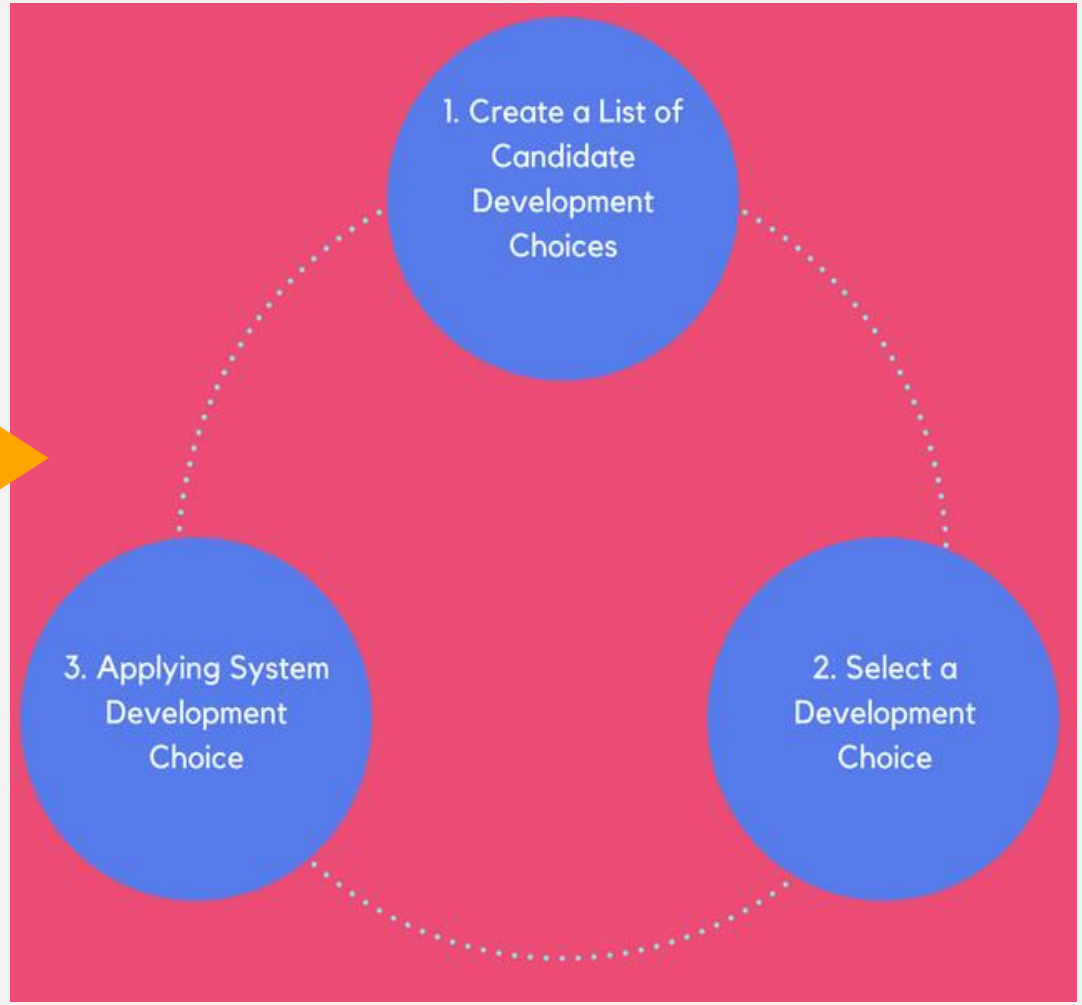
Assurance Based Development assumes:

- the availability of system dependability requirements
- the availability of a description of the *given architecture*



ABD Workflow Overview





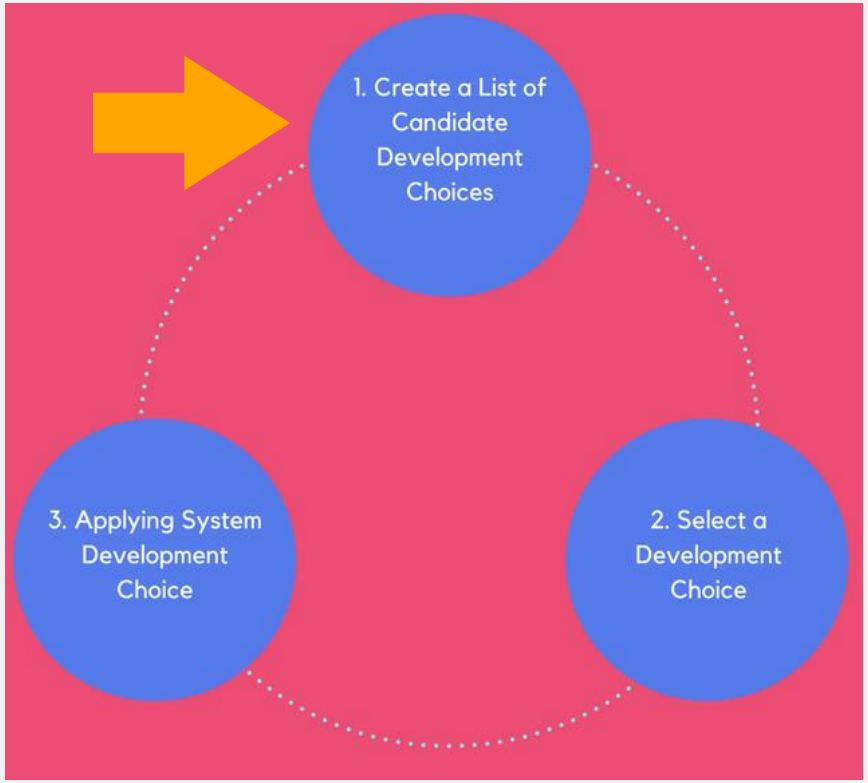
Candidate Development Choices

1. Developers brainstorm choices that will lead to a system that meets its functional, cost, dependability and other goals.

2. Developers enumerates candidate development choices.

3. Developers then consider familiar choices or may solicit suggestions from colleagues.

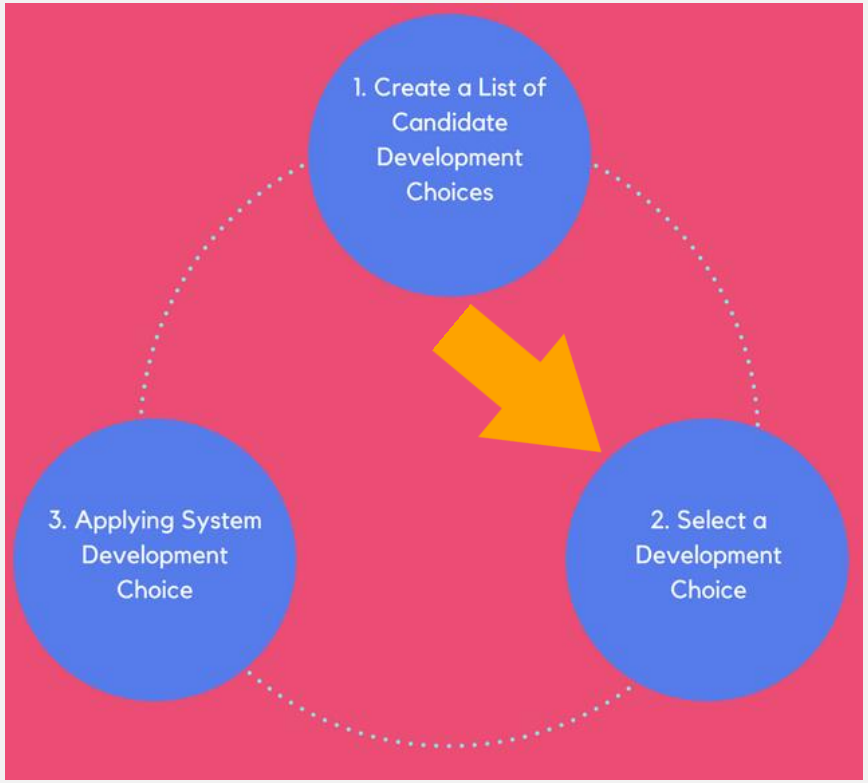
There are costs associated with the consideration of more choices!



Selection of a Development Choice

Selection of a choice is based on 7 criteria:

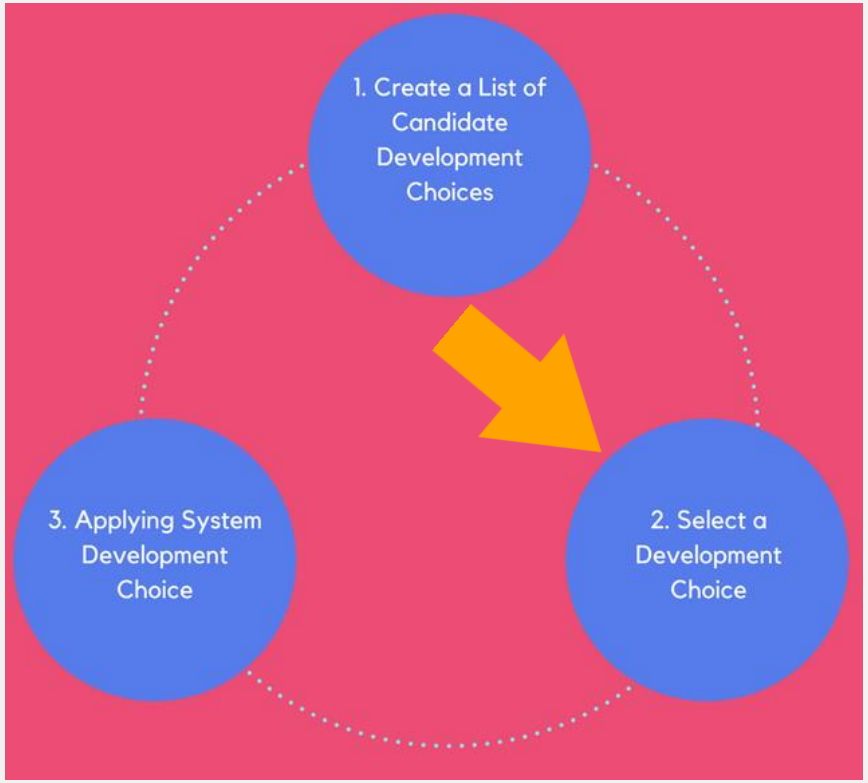
1. *Functionality*
2. *Restriction on later choices*
3. *Evidence of dependability*
4. *Cost*
5. *Feasibility*
6. *Applicable standards*
7. *Non-functional requirements*



Selection of a Development Choice

Example - Anti-lock braking system:

a) A single processor.
b) Two processors whose outputs are compared.
c) Three processors whose outputs will be voted on (TMR).
d) Many processors on a real-time bus.

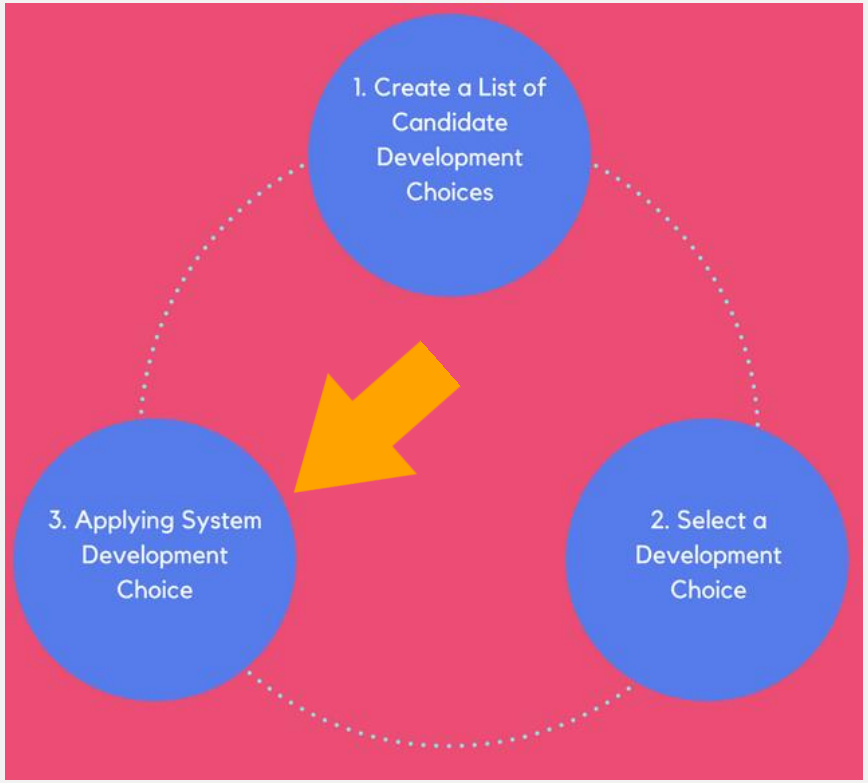


Applying a Development Choice

Once a development choice is made:

1. The choice is applied to the system.

2. The assurance case is updated to reflect its effect.

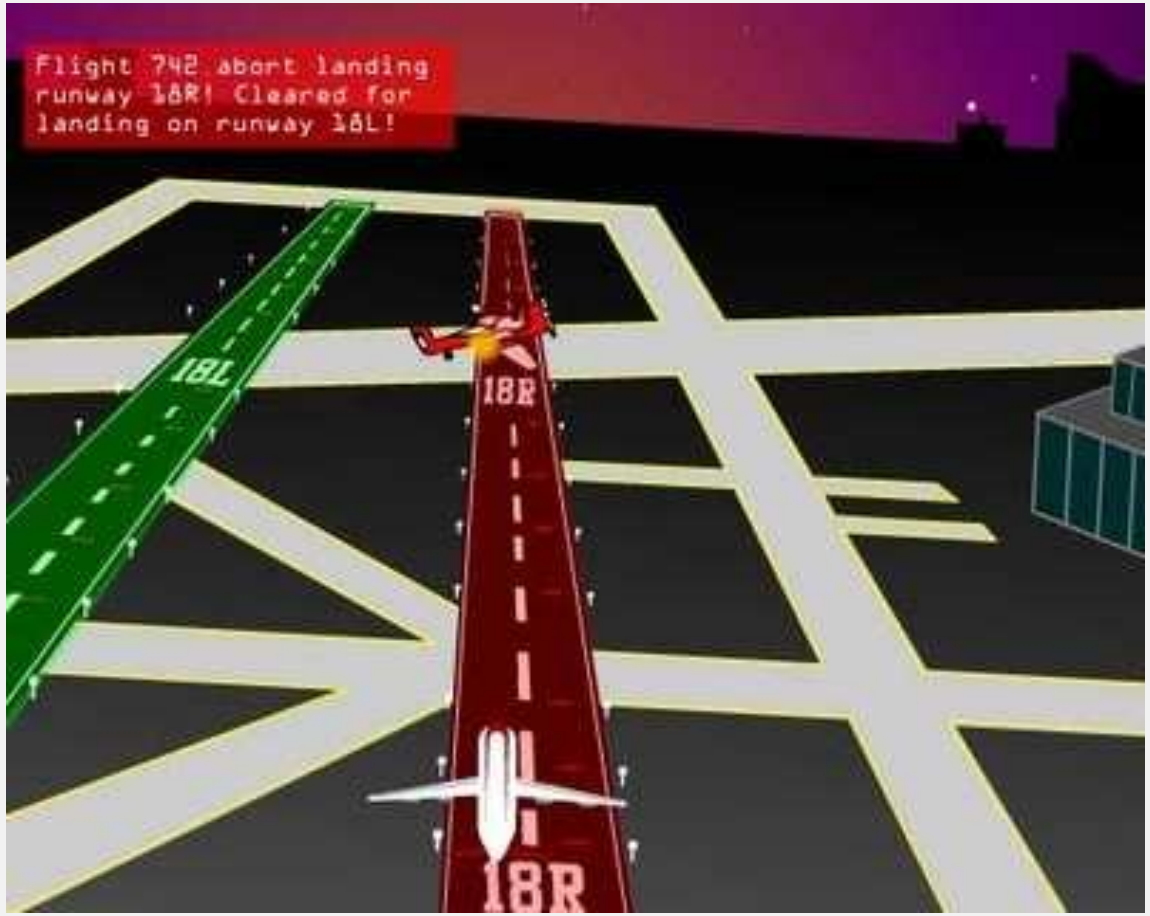


ABD Example

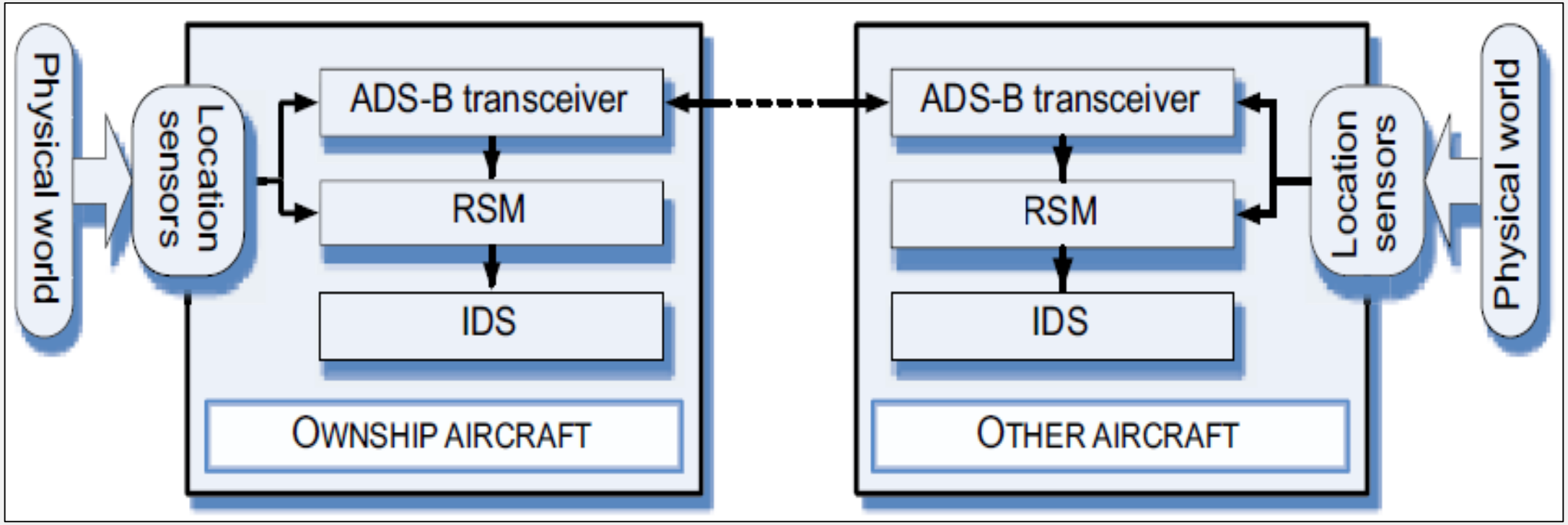
Example system – Runway Incursion Prevention System (RIPS)

- Alerts pilots about potential runway incursions via IDS (Integrated Display System)
- Project developed for NASA

The authors focus on a subcomponent of RIPS, called the Runway Safety Monitor (RSM).



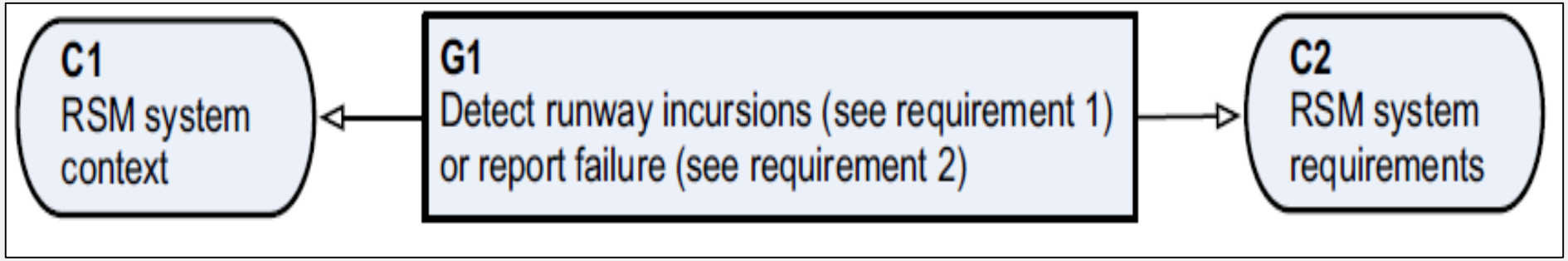
The Given Architecture



Top Level Assurance Goal

Assume that RSM is required to meet the following two requirements:

- If the quality of the supplied data is adequate, detect runway incursions involving ownership within ***t*** time units after they begin with probability greater than or equal to ***P0***.
- If the quality of the supplied data is inadequate, report a failure of RSM with probability greater than or equal to ***P1*** within ***u*** time units.



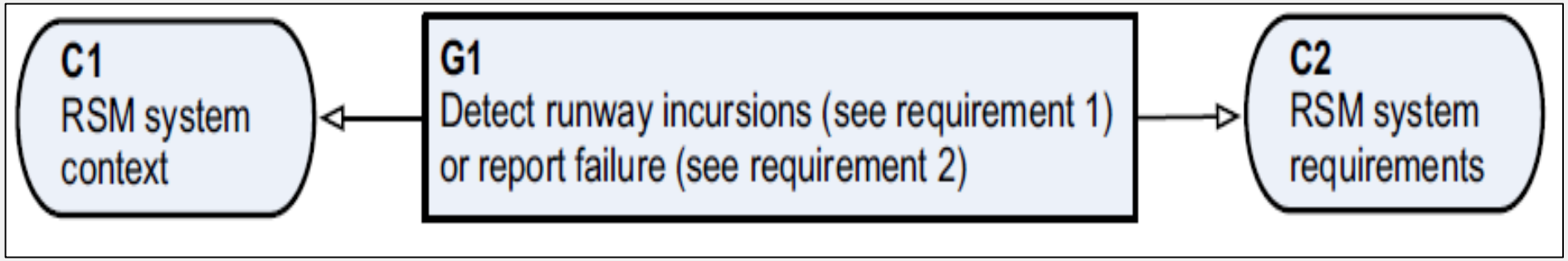
First System Development Choice

Overall approaches for the real-time requirements:

1. Sequential
2. Concurrent
   - Synchronous
   - Asynchronous

Requirement for the detection of corrupt/missing data:

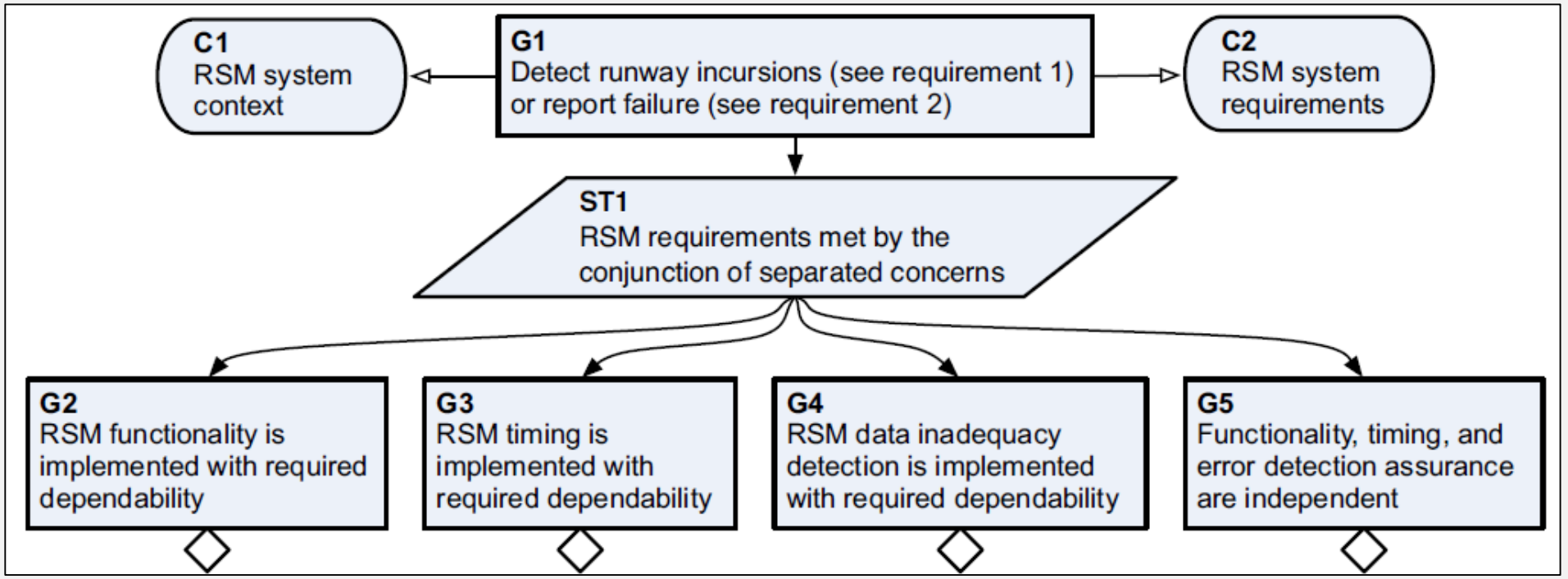
1. A system module can
   - Generate an event
   - Time-out
2. Other



First System Development Choice

Development Choices Made:

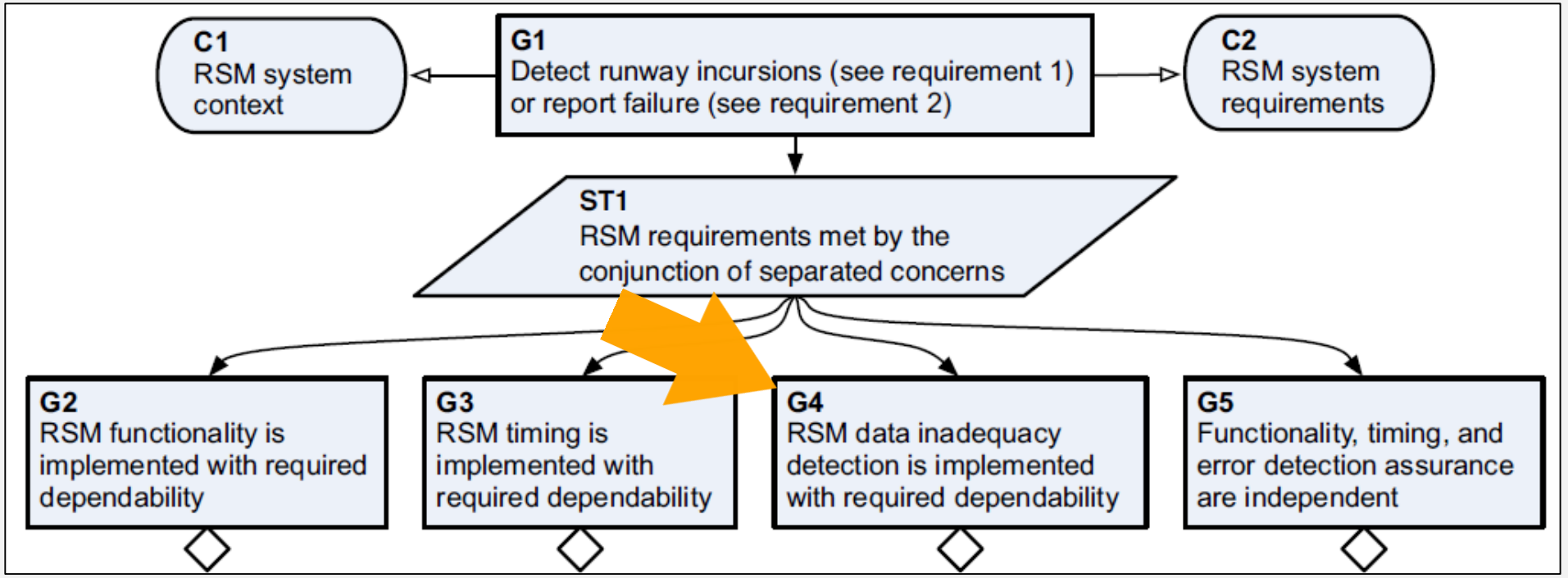
- Sequential code implementation
- Each software module is responsible for detecting and reporting errors in the data that it handles



Second System Development Choice

Available choices to address G4 (failure detection):

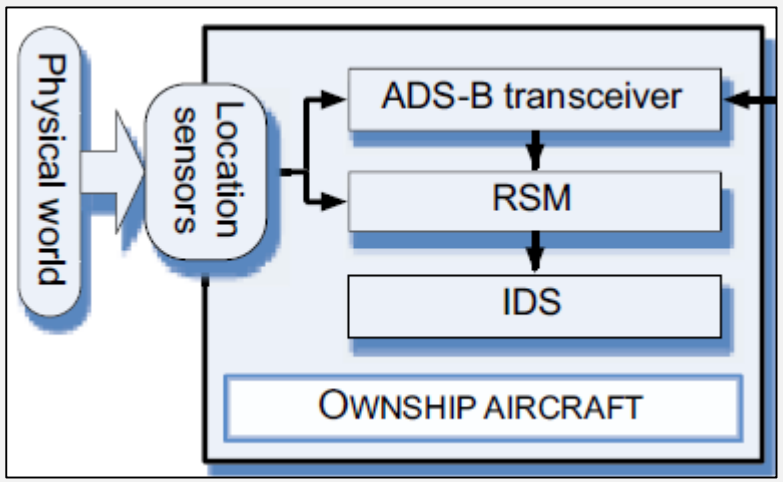
- New architectural pattern
- Implementing an object-oriented architecture
- Functional decomposition

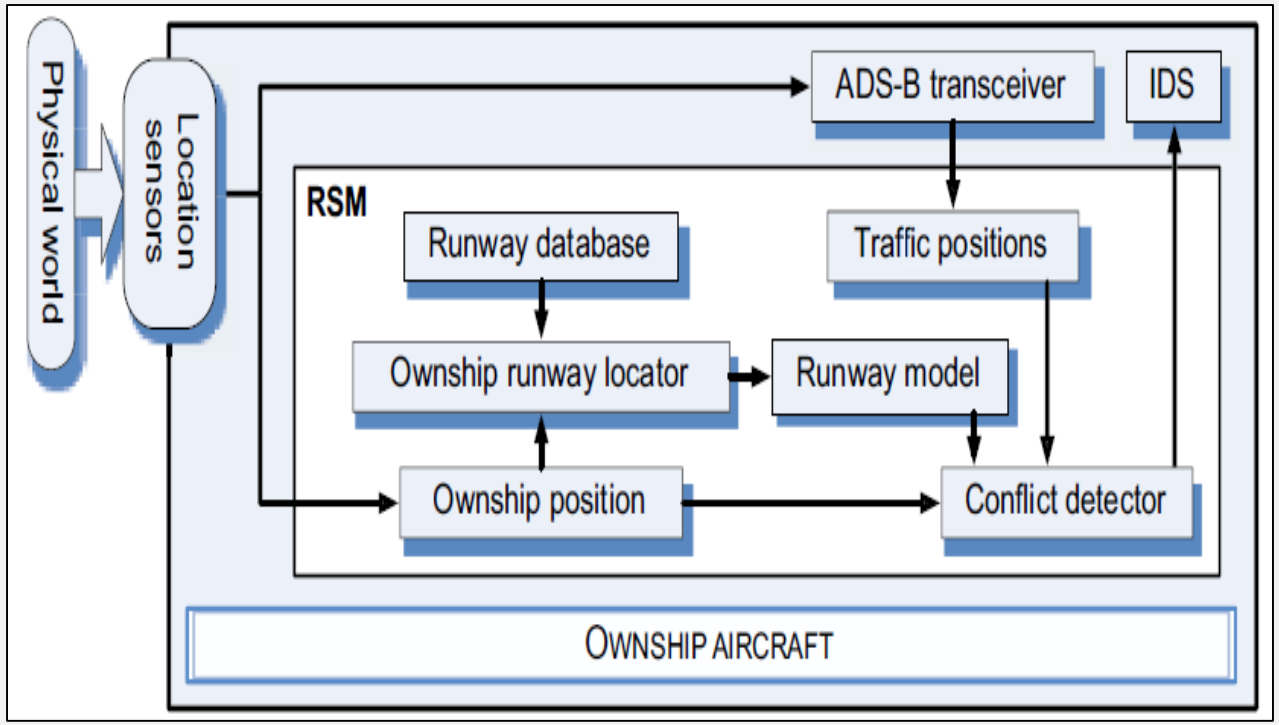


Second System Development Choice

Available choices to address G4 (failure detection):

- New architectural pattern
- Implementing an object-oriented architecture
- Functional decomposition

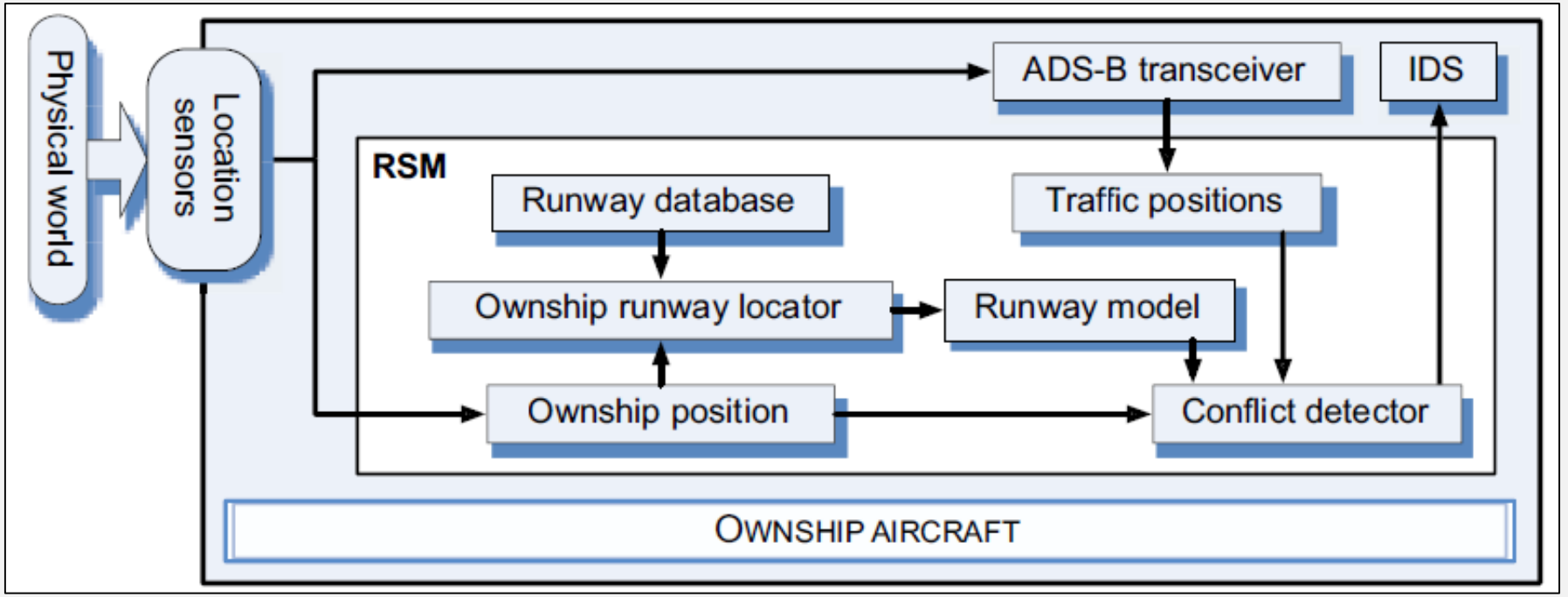




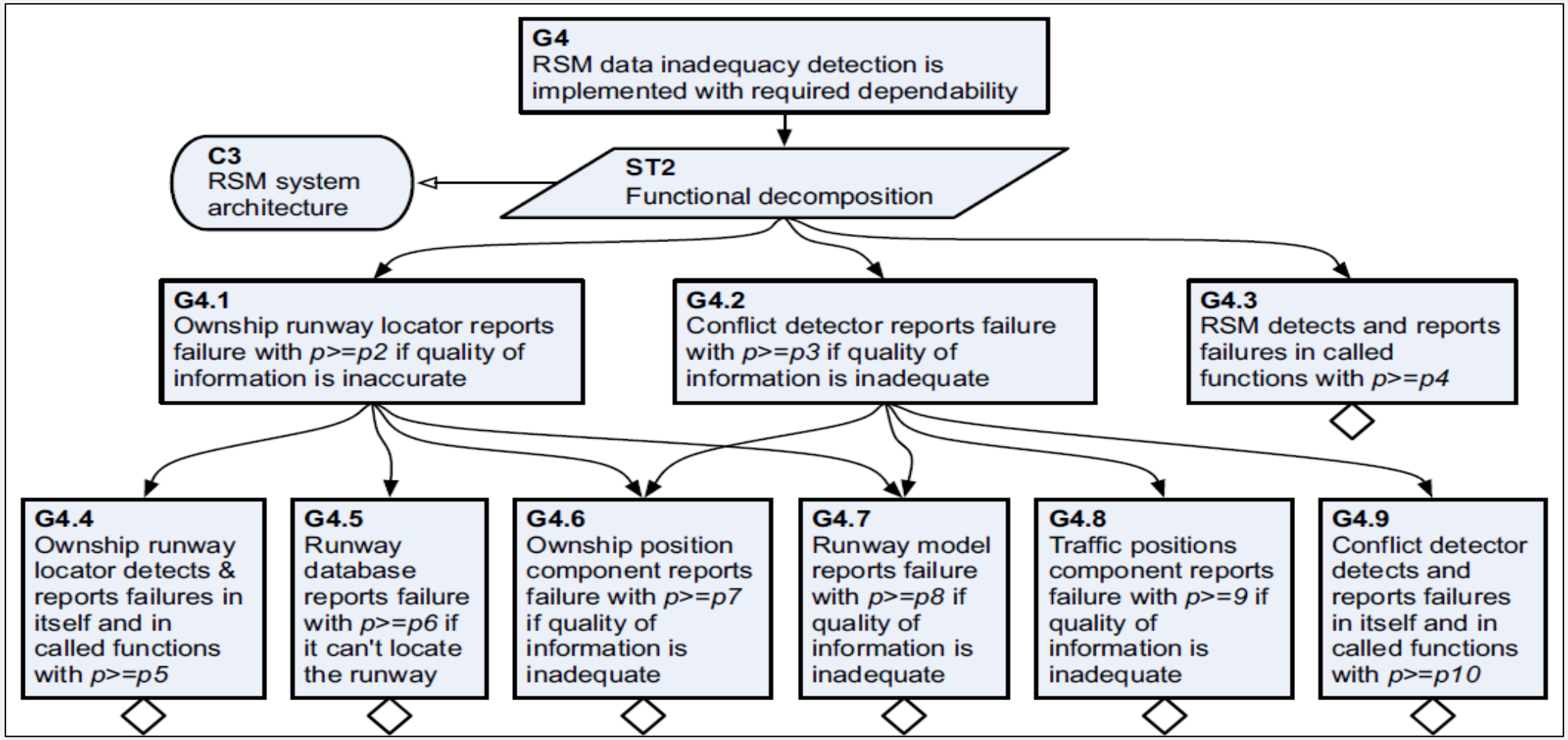
Second System Development Choice

Available choices to address G4 (failure detection):

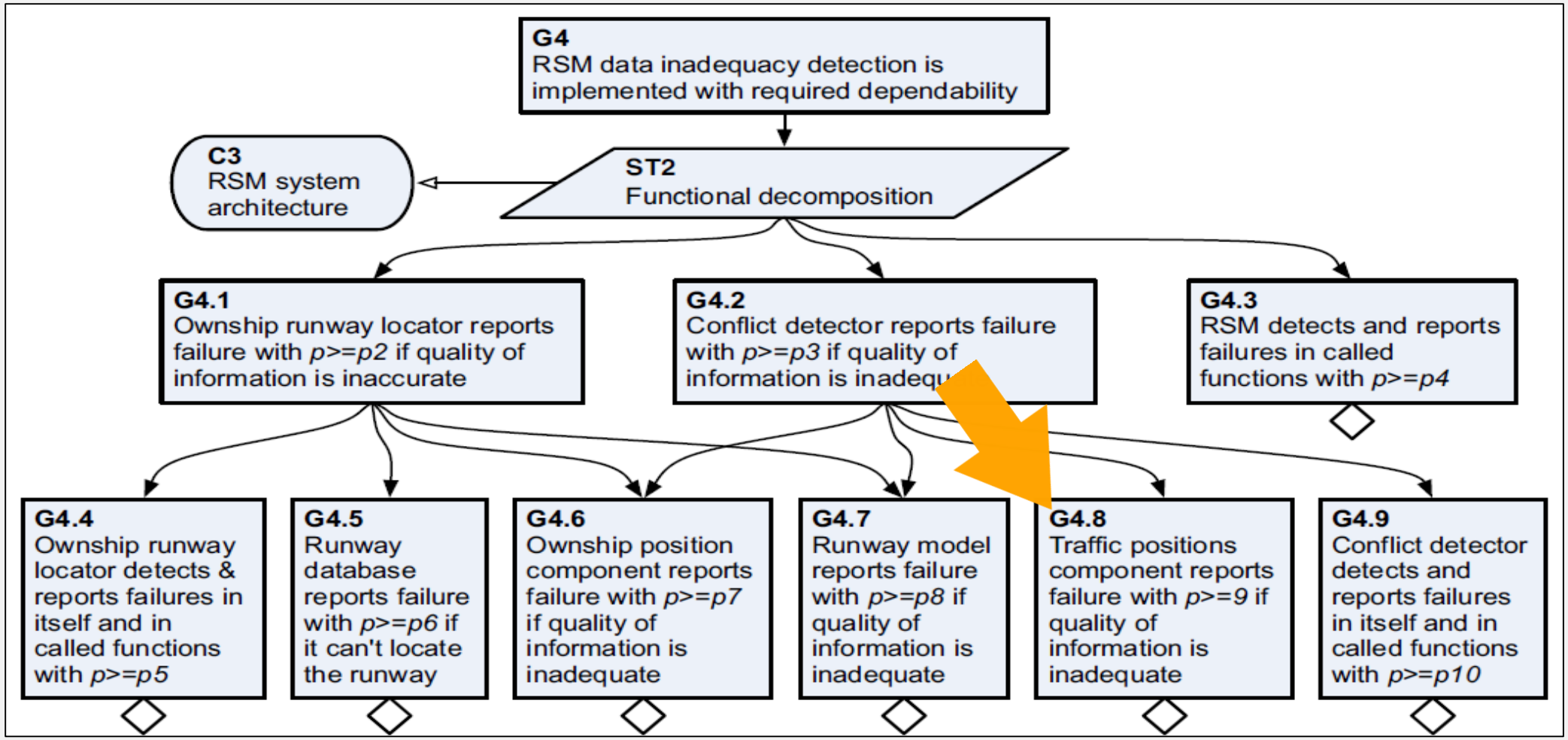
- New architectural pattern
- Implementing an object-oriented architecture
- Functional decomposition



Second System Development Choice

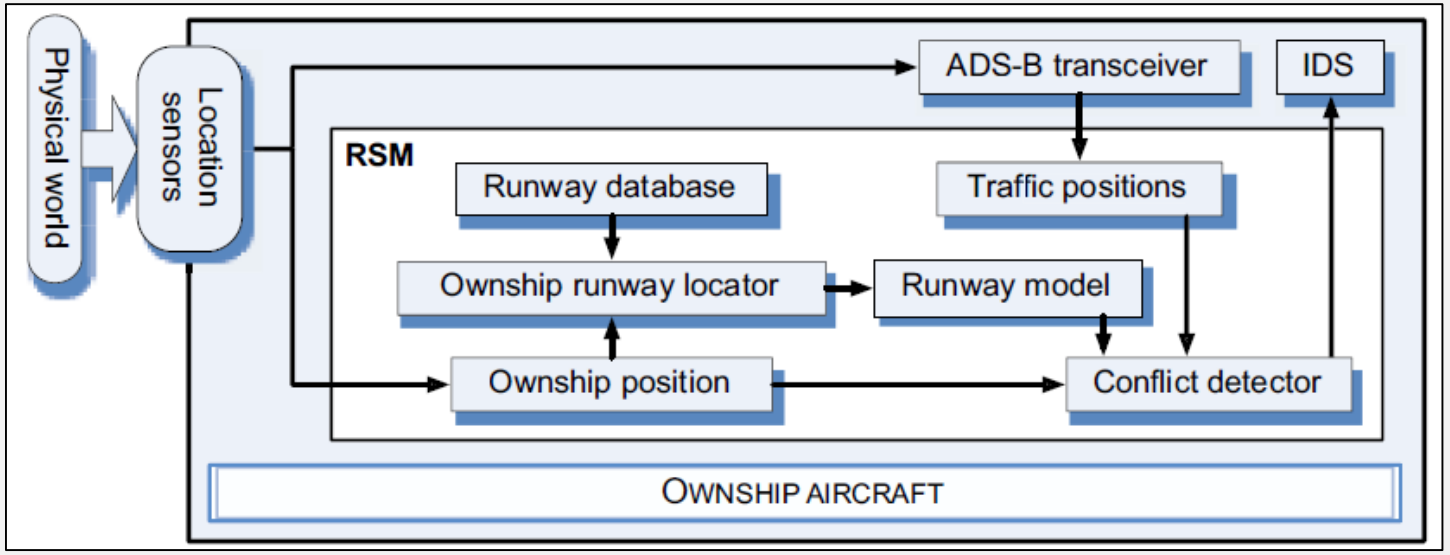


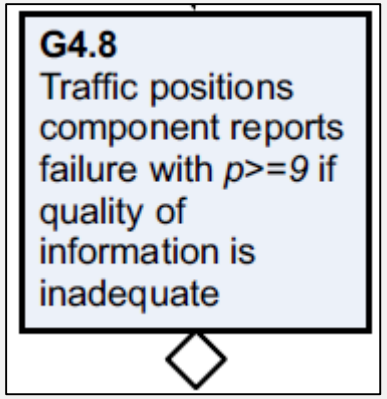
Third System Development Choice



Third System Development Choice

- TPC must detect inadequate information received from ADS-B due to:
  - Other aircraft reporting incorrect data.
  - Data can be corrupted in transit.
  - Data can be stale due to no updated data received

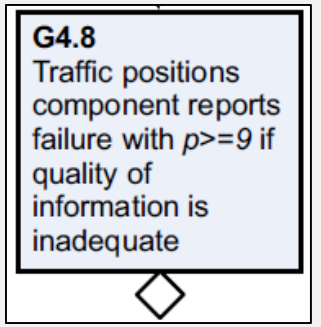




Third System Development Choice

Available choices to address G4.8 :

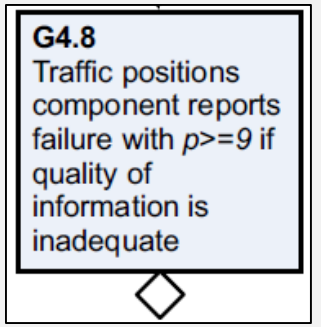
- Impose reasonableness criteria.
- Incorporate redundant source of information, such as a radar or a camera with which to compare information.

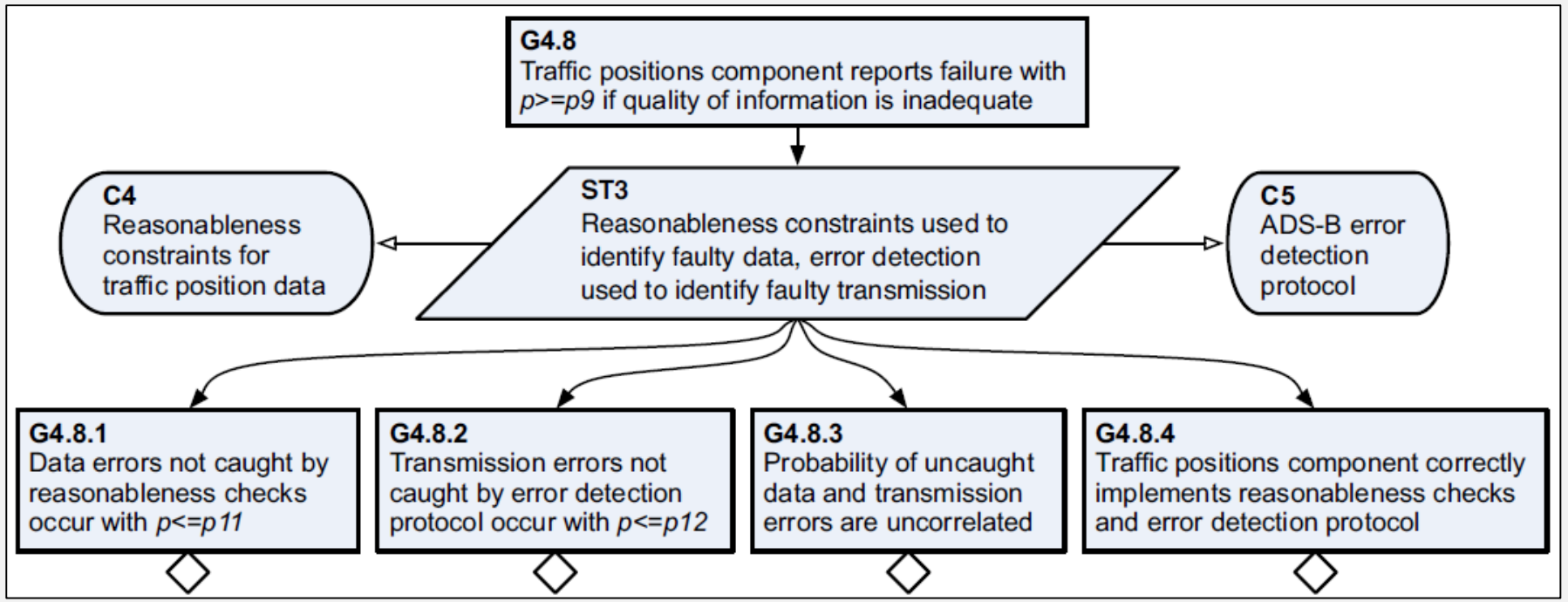


Third System Development Choice

Available choices to address G4.8 :

- Impose reasonableness criteria.
- Incorporate redundant source of information, such as a radar or a camera with which to compare information.

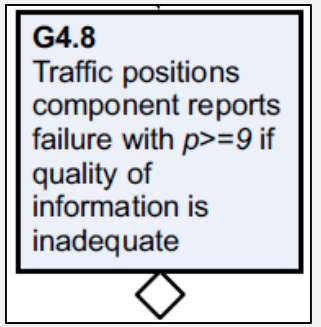


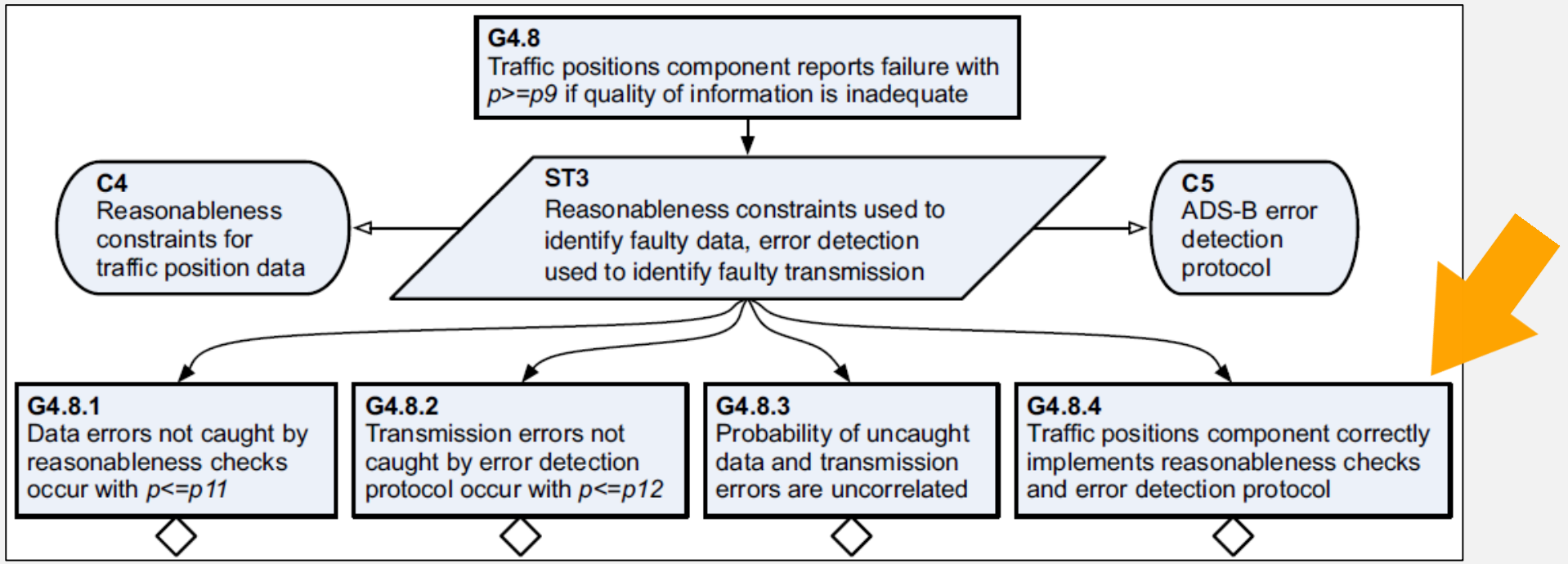


Third System Development Choice

Available choices to address G4.8 :

- Impose reasonableness criteria.
- Incorporate redundant source of information, such as a radar or a camera with which to compare information.

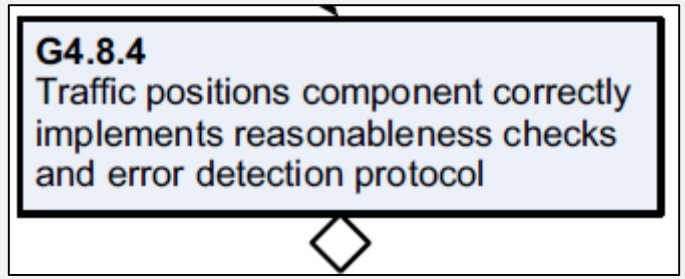




Fourth System Development Choice

Easiest choice to address G4.8.4 :

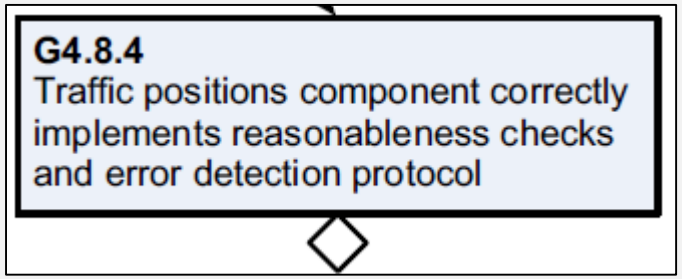
- Use a fully verified implementation of the traffic position component.

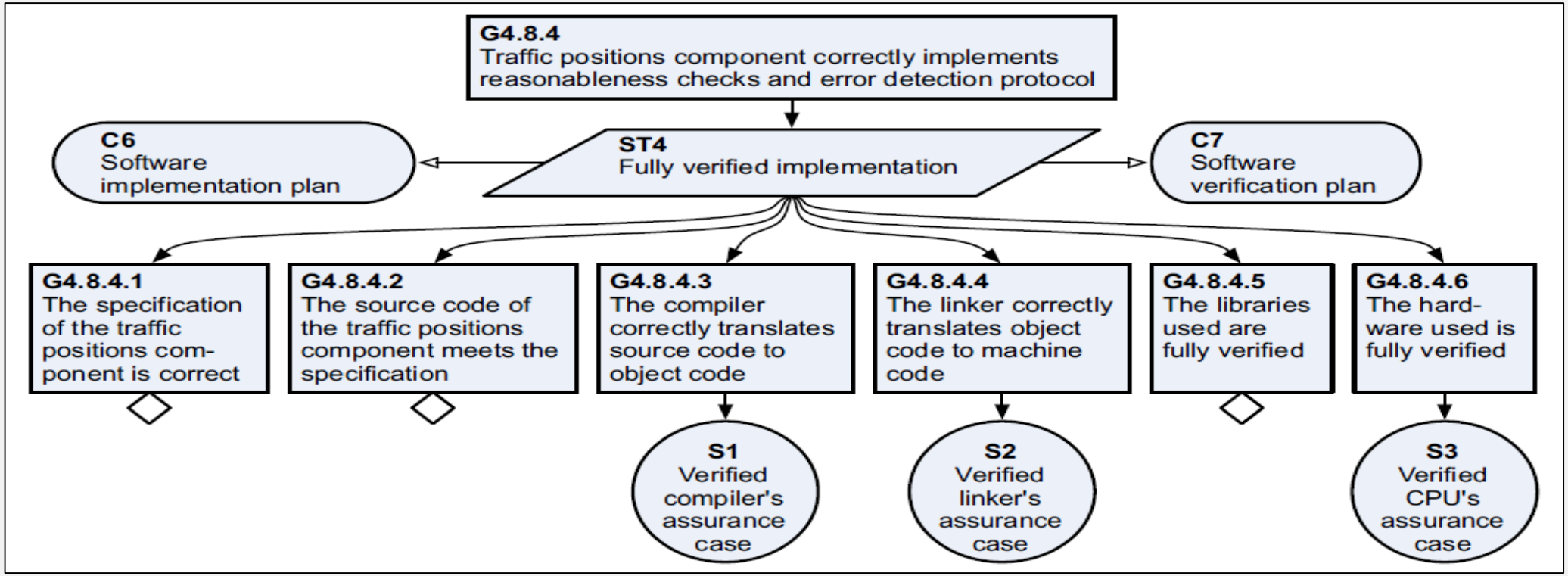


Fourth System Development Choice

Easiest choice to address G4.8.4 :

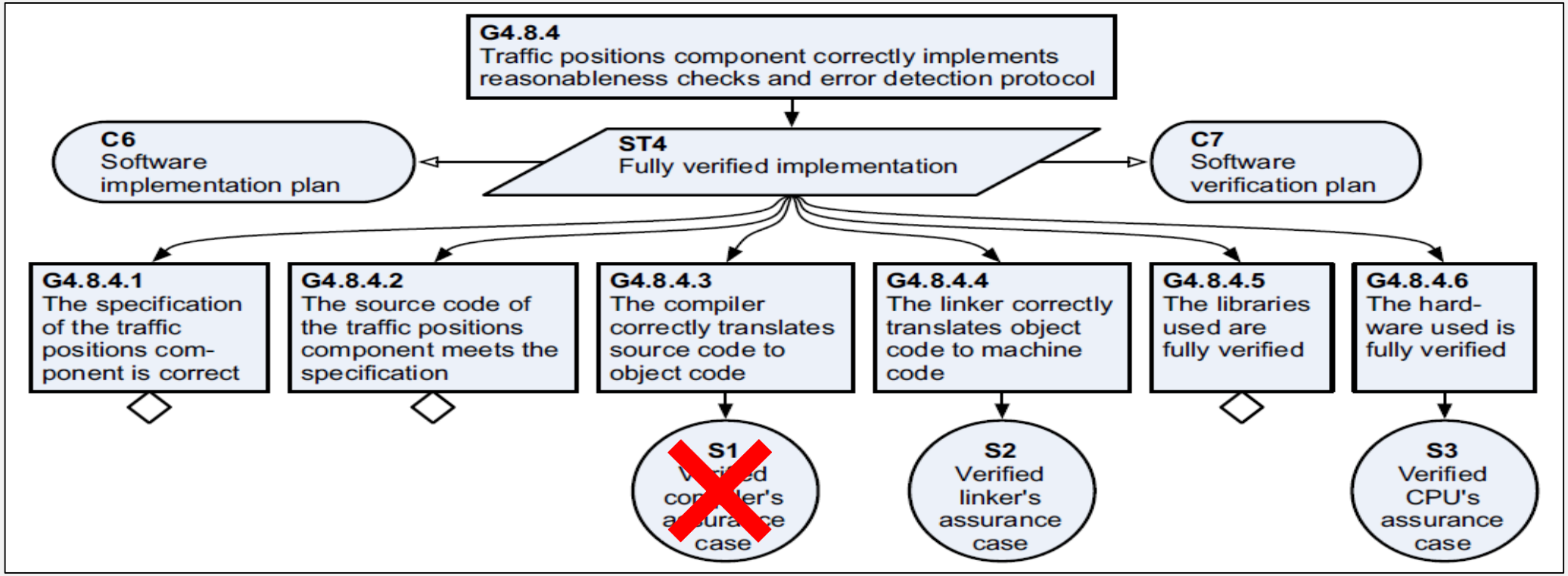
- Use a fully verified implementation of the traffic position component.





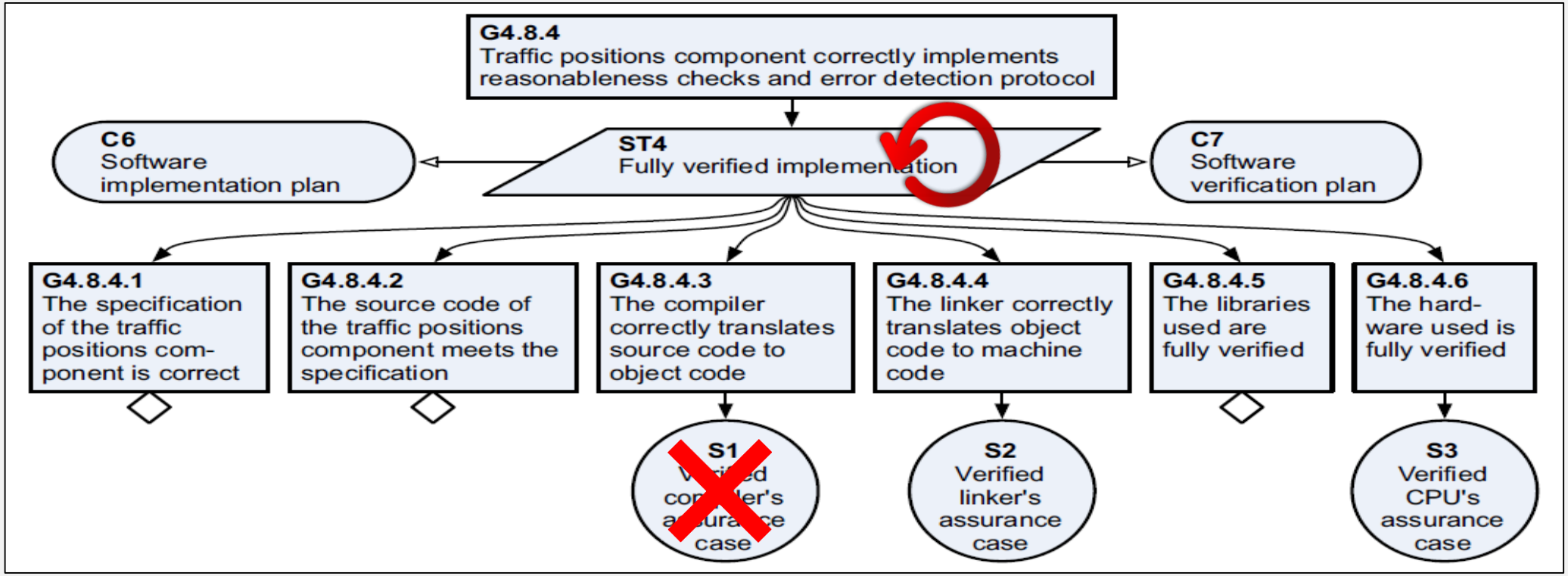
Re-addressing a Choice

- At any point in the process, a developer may discover that a previous choice leads to an unsatisfiable goal.



Re-addressing a Choice

- Then it might be necessary to re-address our previous choice.



Questions

1. Do you foresee any (development) costs that may be associated with using the Assurance Based Development approach?

2. ABD assumes the availability of system requirements, including functional requirements and dependability requirements, as well as the high-level architecture in which the system will operate. Do you believe this is reasonable?

3. Do you think development creativity might be impacted by strictly following the safety case feedback during each development decision? (I.e. The product is dictated by the safety case, not the safety case dictated by the product.)

4. General thoughts about the paper?