MES
MODEL ENGINEERING SOLUTIONS

MIGRATION STRATEGY FOR REVISION ISO 26262:2018 IN SOFTWARE DEVELOPMENT

MGIGroup, 5th March 2019

SOFTWARE QUALITY.
MADE IN GERMANY.

SOLUTIONS FOR INTEGRATED QUALITY ASSURANCE
OF EMBEDDED SOFTWARE

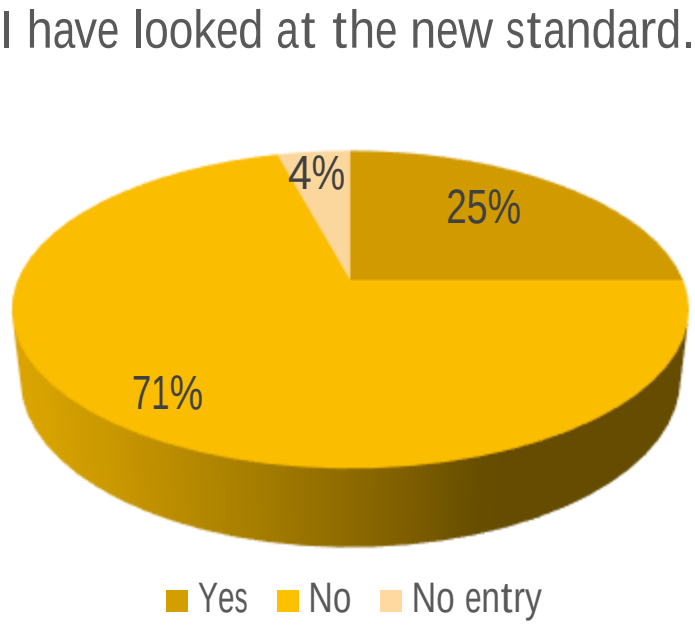
ORGANIZATIONAL INFO

- Questions?
  - Questions and an open discussion are welcome at any time
- You will receive the presentation afterwards by email

MOTIVATION

- Release ISO 26262:2011
  - Major milestone for safeguarding safety-related systems for Road vehicles
  - ISO 26262 references model-based SW development
- Second edition ISO 26262:2018
  - Distributed for review end of 2016
  - Final publication December 2018
  - Captures experiences from the last few years

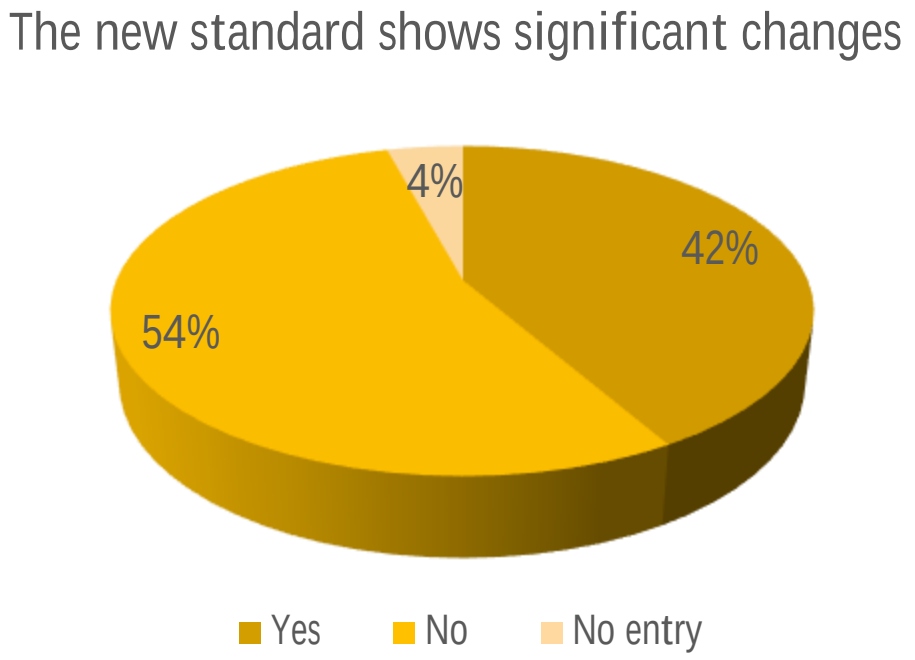
HAVE YOU ALREADY LOOKED AT THE ISO REVISION?



Your Feedback

48 answers as of 05.03.2019

ARE THERE SIGNIFICANT CHANGES?



Your Feedback

48 answers as of 05.03.2019

OUTLINE

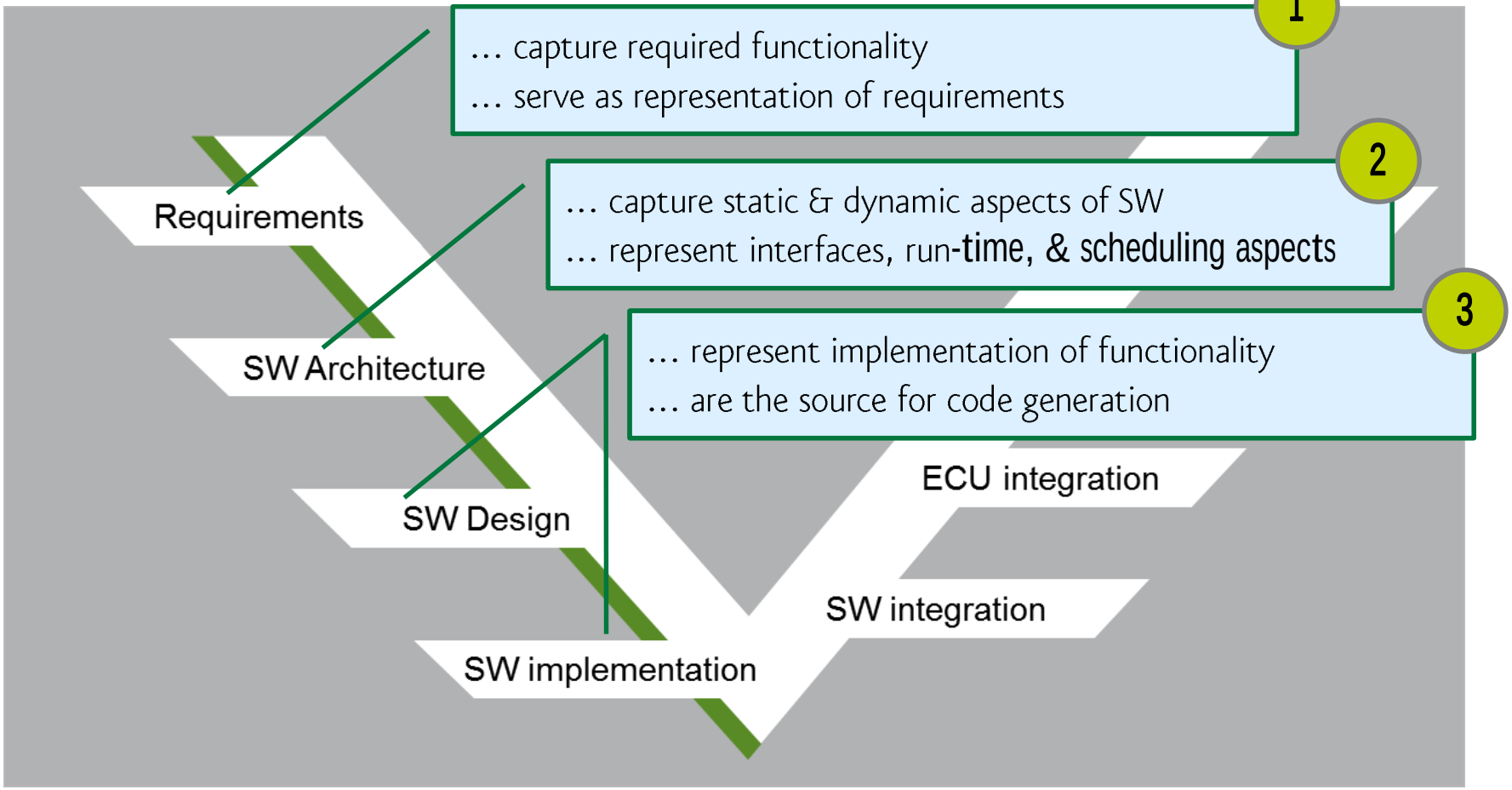
- Use cases of model-based development in Annex B of ISO 26262:2018
- General Updates in ISO 26262:2018
- Changes in individual phases
  - From “Specification of software safety requirements” to “Verification of the software”
- Summary

USE CASES OF MODEL-BASED DEVELOPMENT

- Significant change in Annex B (informative) on model-based development
- Five use cases for model-based software development
  - Specification of software safety requirements
  - Development of the software architectural design
  - Design and implementation of software units, with or without automated code generation
  - Design, implementation and integration of software components, including automated code generation from software component models
  - Verification (static and/or dynamic)
- Benefit
  - Unified reference for communication
  - Clarification: e.g. models for SW safety requirements might not be used for integration of software components

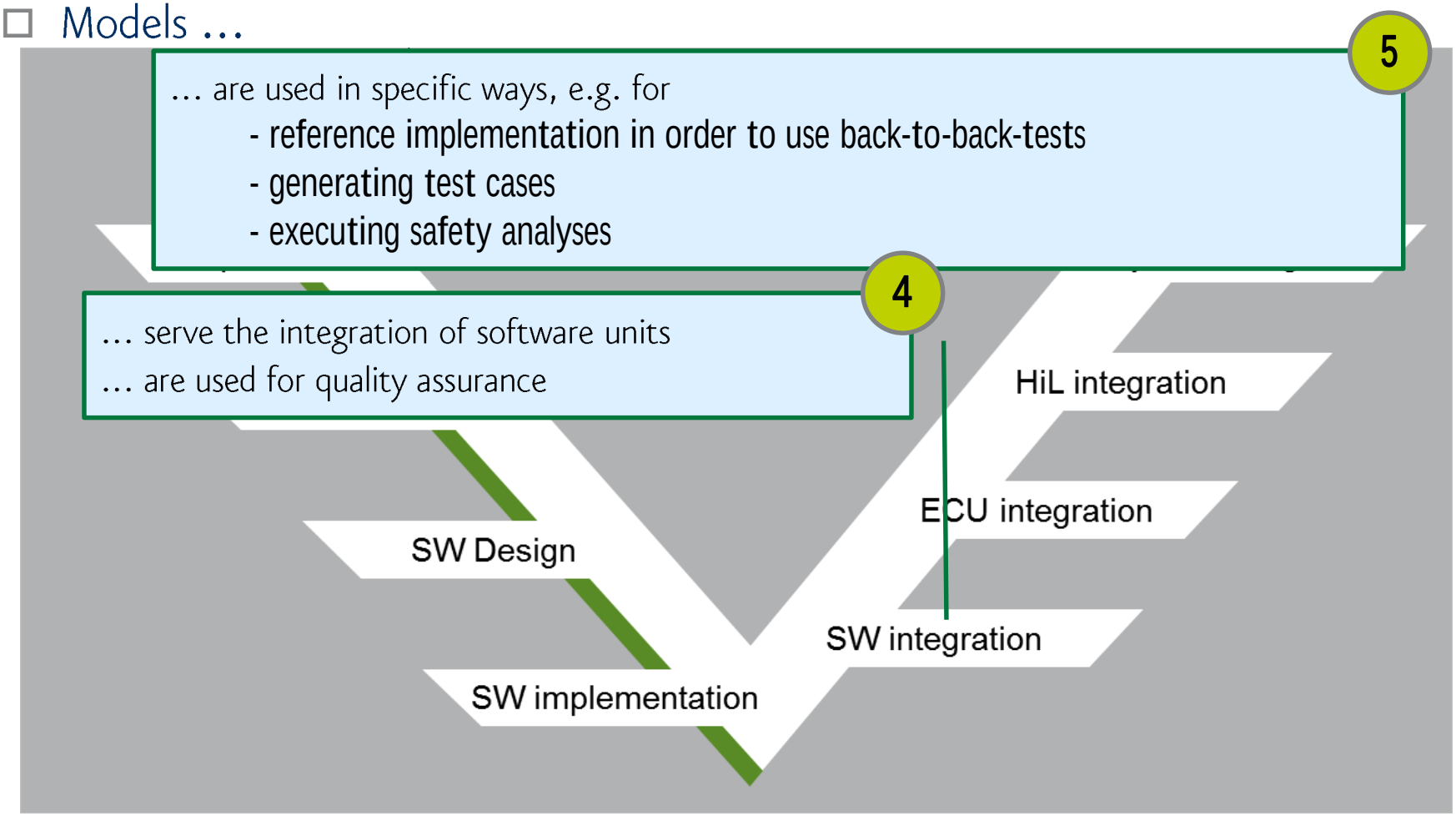
USE CASES OF MODEL-BASED DEVELOPMENT

- Models …



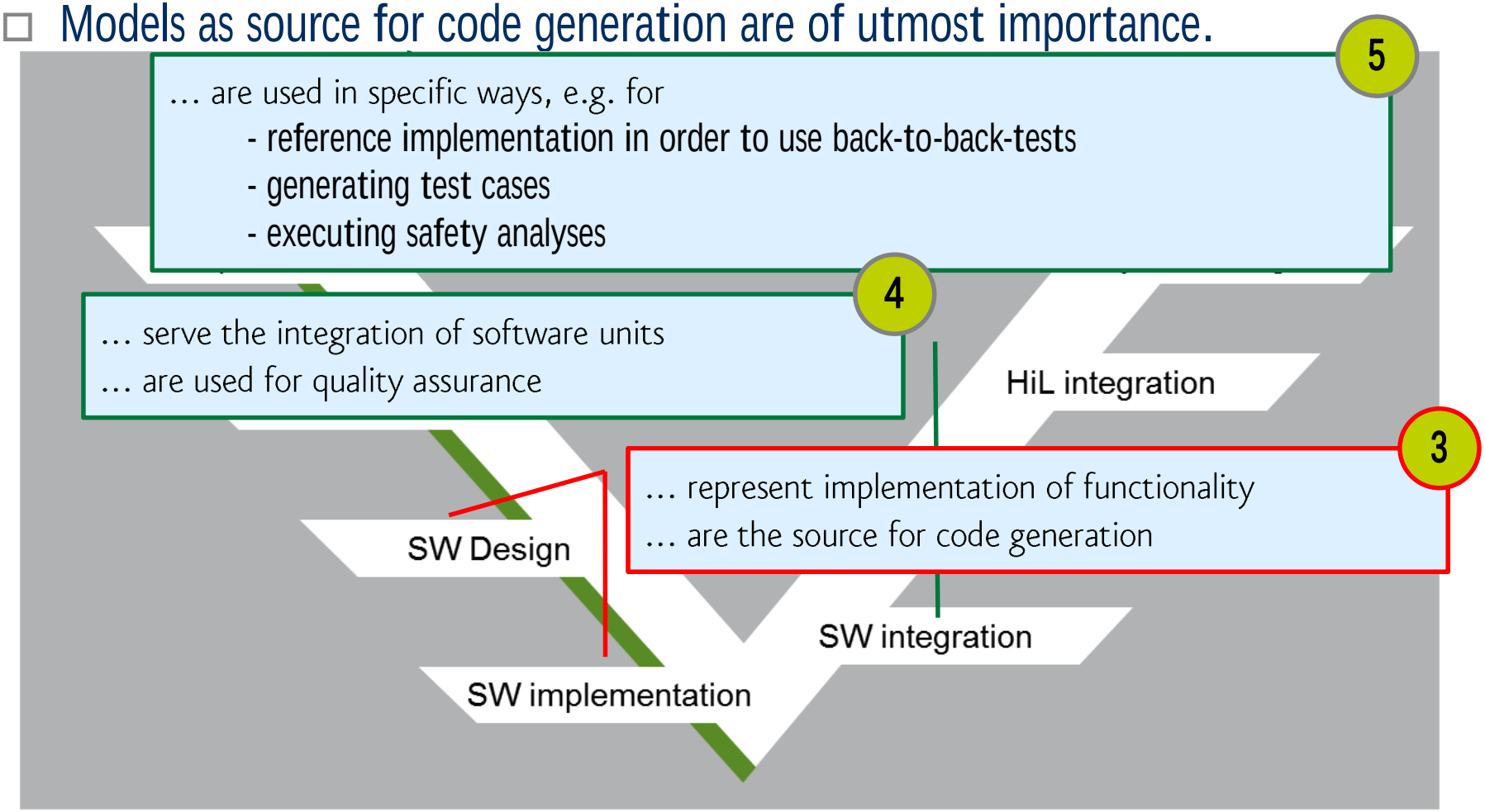
USE CASES OF MODEL-BASED DEVELOPMENT

- Models …



USE CASES OF MODEL-BASED DEVELOPMENT

- Models as source for code generation are of utmost importance.



OUTLINE

- Use cases of model-based development in Annex B of ISO 26262:2018
- General Updates in ISO 26262:2018
- Changes in individual phases
  - From “Specification of software safety requirements” to “Verification of the software”
- Summary

TYPES OF CHANGES

- Tables transformed to notes, (no longer recommendations for ASIL levels)
- Changed rating of recommendation (mostly stricter)
- New methods/topics/principles

- New methods are listed that address typical issues arising from concurrent software execution on multi-core systems

- Extensions by methods:
  - Table 1 "Modelling and Coding guidelines" asks for guidelines on the "Representation of concurrency aspects"
  - 7.4.12. Note 2 (former Table 4) "Mechanisms for error detection" addresses "Temporal monitoring of program execution" and "Access violation control mechanisms"
  - Table 4 (former Table 6) "Methods for the verification of the software architectural design" asks for "scheduling analysis"

Did you already prepare the process to address concurrency aspects in the quality assurance measures? And if, how?

EVOLUTION OF TECHNOLOGY AND BEST PRACTICES

- 7.4.12 Note 3 (former Table 5) "Mechanisms for error handling": more detailed mechanisms for redundancy
  - "Independent parallel redundancy" split into recommendations on a) homogeneous and b) diverse redundancy in the design
- Table 7 (former Table 9) "Methods for software unit verification": added verification techniques
  - Pair-programming
  - Static analyses based on abstract interpretation

ISO emphasizes additional state of the art methods like pair programming or intensified static analysis.
Which obstacles do you see realizing those methods in daily business?

EVOLUTION OF TECHNOLOGY AND BEST PRACTICES

- Table 10 (former Table 13) "Methods for verification of software integration" added verification techniques
  - No longer just applicable to software units but also to software integration
  - E.g. analyses on control or data flow, "static code analysis", as well as "static analyses based on abstract interpretation"
- New table 15 "Methods for deriving test cases" for testing embedded software
  - Recommended methods partially drawn from verification of software integration
  - Added analysis of functional dependencies and operational use cases

EVOLUTION OF TECHNOLOGY AND BEST PRACTICES

- The method recommendation level has been adopted in various tables
  - Reflects the evolved state-of-the-art nature
  - Several techniques have been proven to be more powerful

    ➔ Promotion from a simple "recommendation", i.e. "+", to "highly recommended", i.e. "++"
  - Recommendation of other techniques or methods has been relaxed

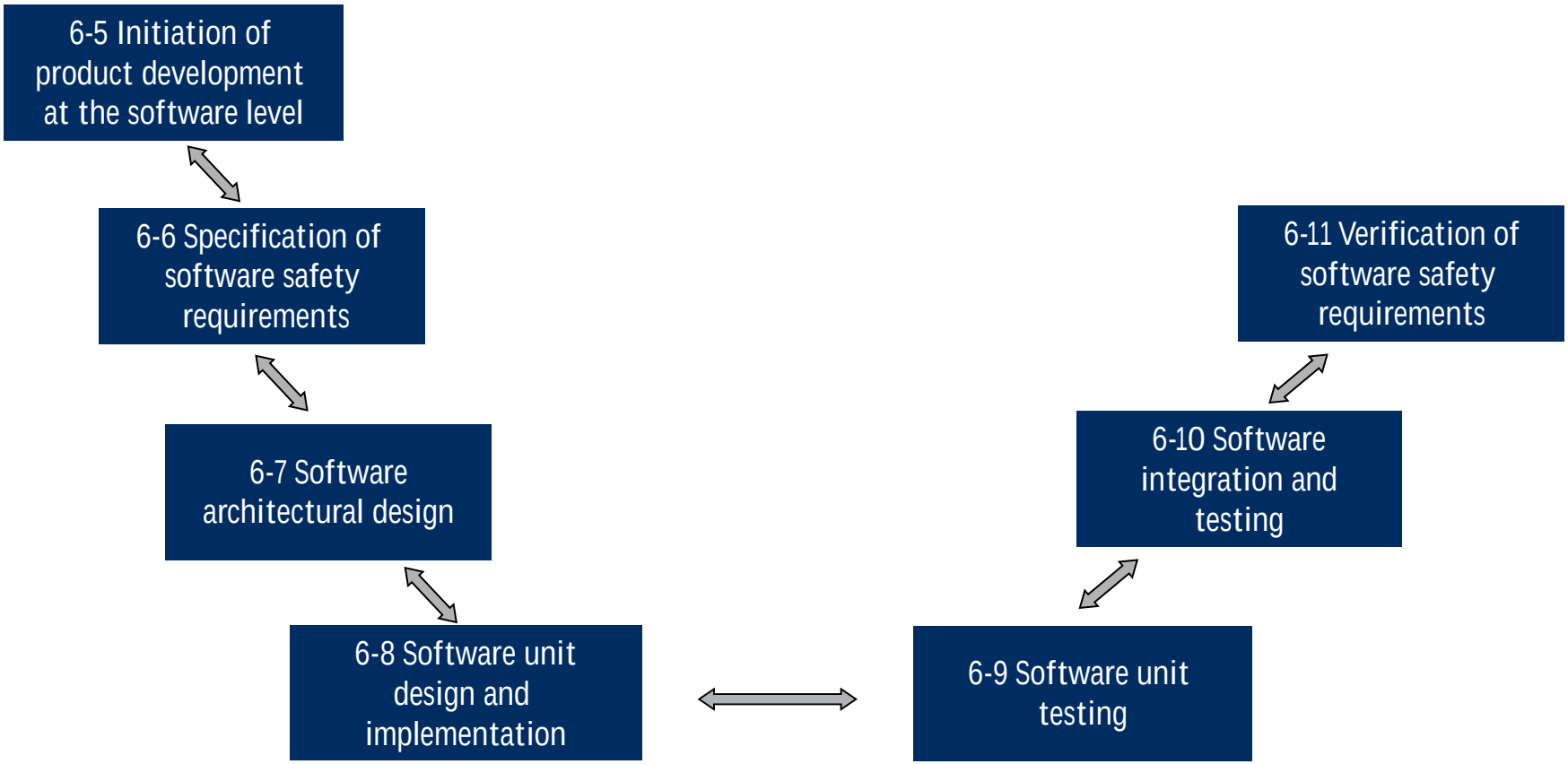
		2011	2018
5 Notations for software unit design	1c) Semi-formal notations	ASIL B: ++	ASIL B: +
6 Design principles for software unit design	1d) No multiple use of variable names	ASIL A: +	ASIL A: ++
	1f) Restricted use of pointers	ASIL A: o ASIL B: + ASIL C: +	ASIL A: + ASIL B: ++ ASIL C: ++

Change impact?

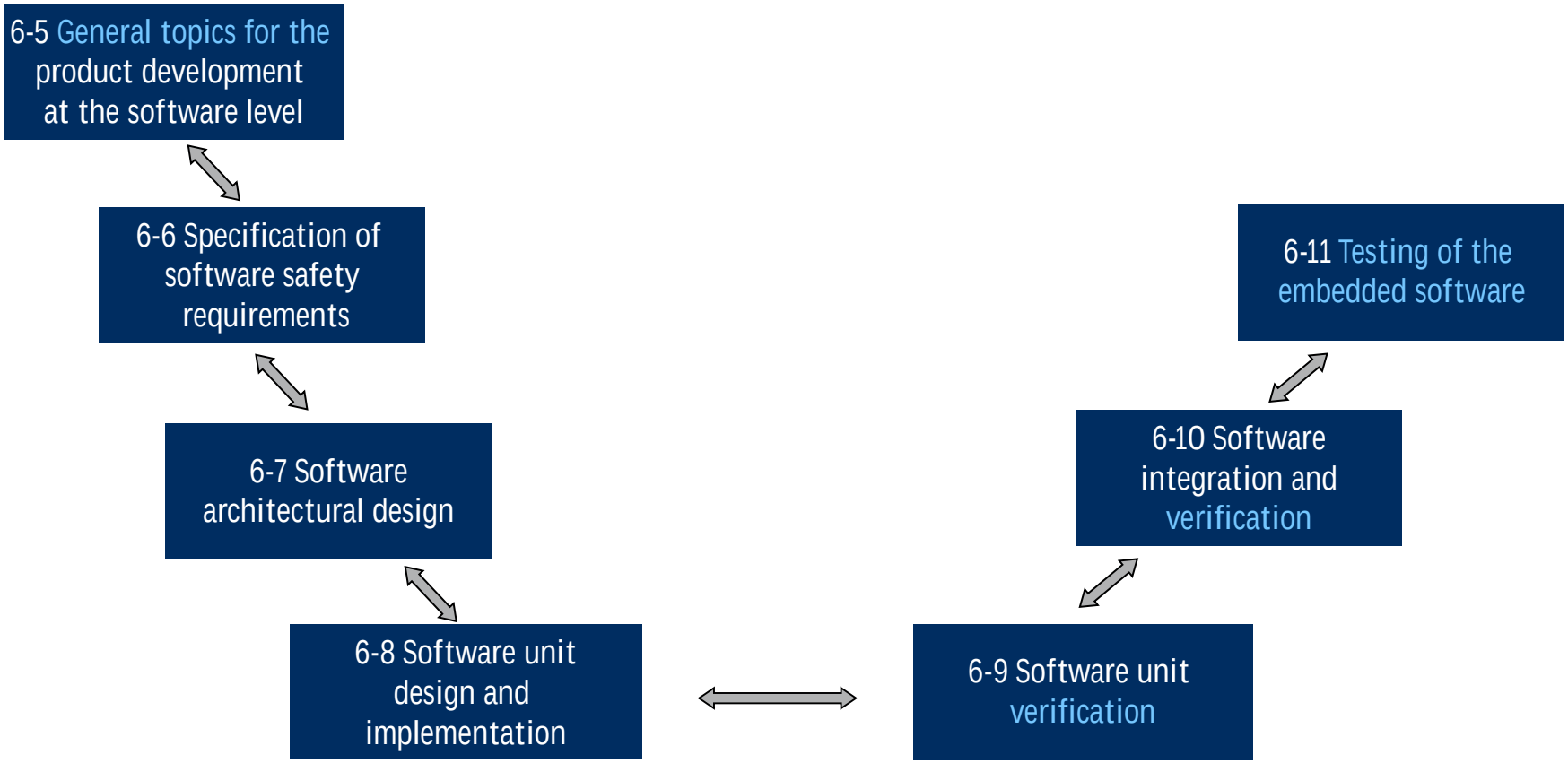
OUTLINE

- Use cases of model-based development in Annex B of ISO 26262:2018
- General Updates in ISO 26262:2018
- Changes in individual phases
  - From “Specification of software safety requirements” to “Verification of the software”
- Summary

REFERENCE PHASE MODEL ISO 26262:2011



REFERENCE PHASE MODEL ISO 26262:2018



What differences are important? E.g., how do testing and verification differ?

Will these changes have effects on your process?

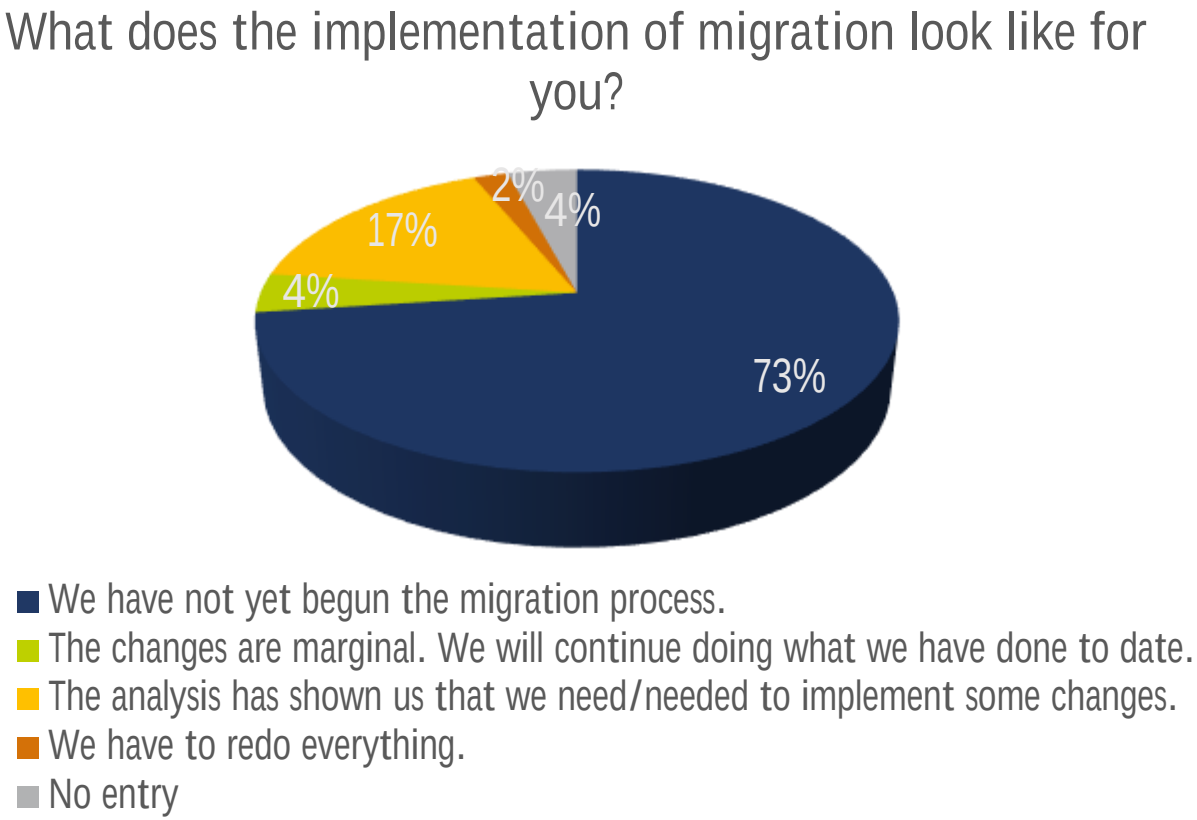
OUTLINE

- Use cases of model-based development in Annex B of ISO 26262:2018
- General Updates in ISO 26262:2018
- Changes in individual phases
  - From “Specification of software safety requirements” to “Verification of the software”
- Summary

SUMMARY

- Model-based development use cases have been refined
  - "Design and implementation of software units" explicitly mentioned
  - Four further use cases for models in SW development
- General findings
  - Recommendations extended to software development for multi-core HW platforms
  - Evolution of technology and best practices reflected by extension of methods and change of degree of recommendations
- Changes in individual phases
  - Model-based development with subsequent code generation is the standard use case; manual coding from models not recommended
  - Verification methods lifted from unit to integration level, and testing extended by analyses

MIGRATION PLANS



Your Feedback

48 answers as of 05.03.2019

NEXT STEPS PLANNED

- Standard Approach:
  - „Mature“ projects are implemented accoring to ISO 26262:2011
  - New or early-stage projects are evaluated according to ISO 26262:2018

Your approach and the planned next steps?

MGI GROUP – NEXT ONLINE MEETING

- Tuesday, June 11, 2019
  - 3 p.m. CEST (Berlin)
  - 9 a.m. EDT (Detroit)
  - 6:30 p.m. IST (Bangalore)
  - 9 p.m. CST (Beijing)
  - 10 p.m. JST (Tokyo)

- Registration:
  - https://model-engineers-event.webex.com/model-engineers-event-en/onstage/g.php?MTID=eed9df94ee843402944504ba025496dfa

ADDRESS MAJOR OBSTACLES OF AGILITY

- Upcoming webinar topic:
  - „ISO 26262 in 10 Steps“ (March 12 an 13, 2019)
- Upcoming training:
  - „Model-based Development of Embedded Software in Compliance with ISO 26262 – Challenges and Effective Solutions“ (April 29 – 30, 2019)

SAVE THE DATE!

- MES Summer School 2019
  - June 24 – 28, 2019 | Berlin (Germany)
  - May 3, 10, 17, 31 and June 9, 2019 | Troy (Detroit), MI (U.S.A)
  - September 23 – 27, 2019 | Troy (Detroit), MI (U.S.A.)

- Please find all information on the MES website:
  - https://www.model-engineers.com/mes-summer-school/

MES
MODEL ENGINEERING SOLUTIONS

MODEL ENGINEERING SOLUTIONS GMBH

Waldenserstraße 2 - 4
10551 Berlin
Germany

T: +49 30 2091 6463-0
F: +49 30 2091 6463-33
info@model-engineers.com
www.model-engineers.com

References

(1) ISO 26262-6:2011. Road vehicles -- Functional safety – Part 6: Product development at the software level. ISO/TC 22/SC 3

(2) ISO/DIS 26262-6:2018. Road vehicles -- Functional safety. ISO/TC 22/SC 32

(3) Doerr, H., End, T., and Kaland, L., "On the Impact of the Second Edition of the ISO 26262 on Model-Based Development of Safety-Related Systems," SAE Technical Paper 2017-01-0060, 2017, doi:10.4271/2017-01-0060.