Principles of Feature Modeling

Damir Nešić

KTH Royal Institute of Technology, Stockholm, Sweden
damirn@kth.se

Jacob Krüger

Otto von Guericke University, Magdeburg, Germany
jkrueger@ovgu.de

Ştefan Stănciulescu

ABB Corporate Research, Baden-Dättwil, Switzerland
scas@itu.dk

Thorsten Berger

Chalmers | University of Gothenburg, Gothenburg, Sweden
thorsten.berger@chalmers.se

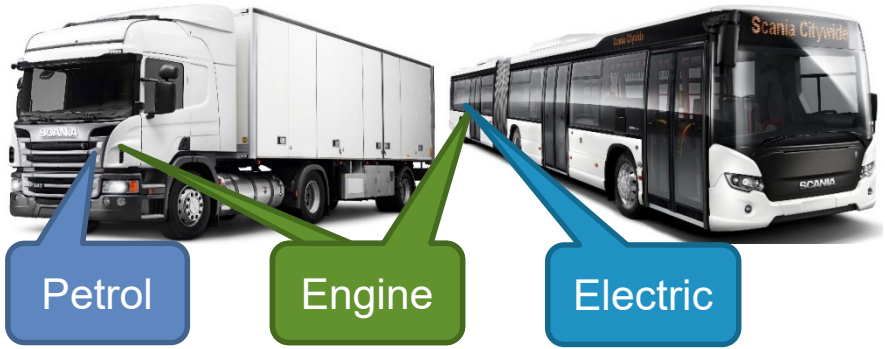
REVaMP²

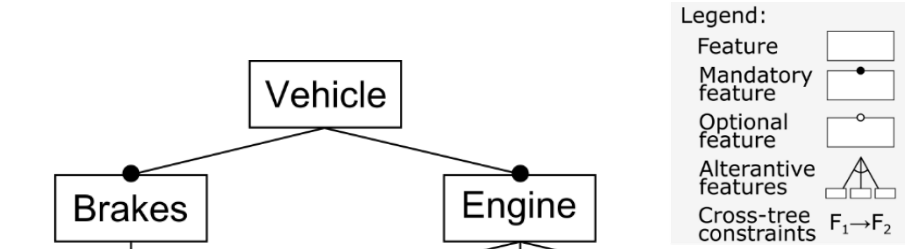
Vetenskapsrådet

PRYSTINE

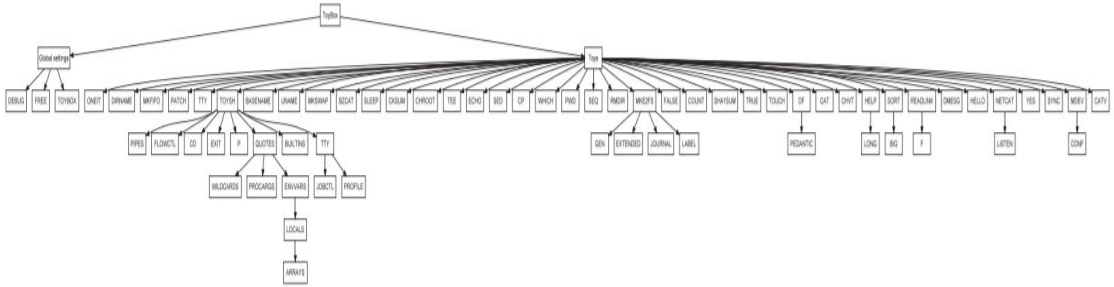
Variability is everywhere

Managing variability





Example feature model



ToyBox feature model with 71 features

30 years of feature-model research

Feature-oriented domain analysis (FODA) feasibility study
KC Kang, SG Cohen, JA Hess, WE Novak, AS Peterson - 1990 - apps.dtic.mil
Cited by 4615 Related articles All 12 versions

Automated **analysis** of **feature** models **20 years** later:
D Benavides, S Segura, A Ruiz-Cortés - Information systems, 2010 - Els
Cited by 1151 Related articles All 11 versions

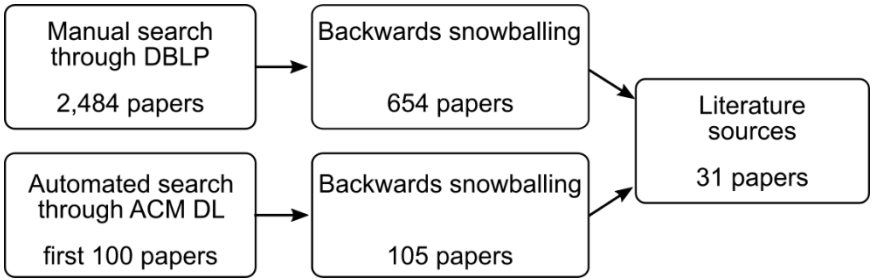
Refactoring

Quality assurance…

- As a step towards a feature-modeling methodology, we contribute:
  - **34 general feature-modeling principles organized into eight categories**
  - **Detailed discussion of each principle with guidelines for their use**

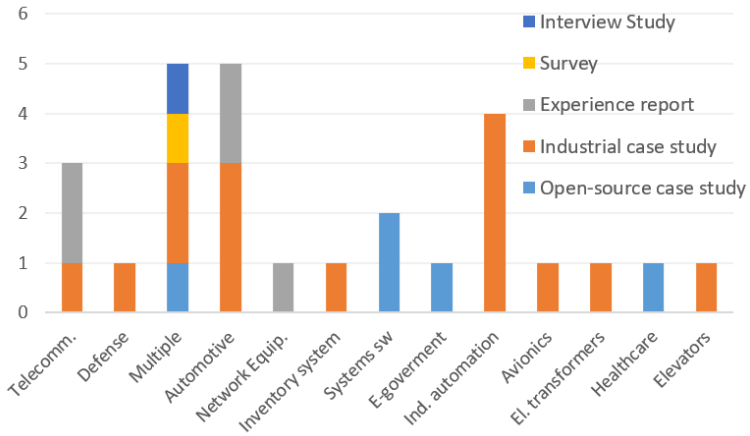
Literature sources

- Analyzed venues: ASE, ESEC/FSE, FOSD, ICSE, ICSME, ICSR, MODELS, SANER, SPLC, and VaMoS.
- Included empirical studies and experiences from practitioners, tool-vendors, educators



Methodology for the identification literature sources

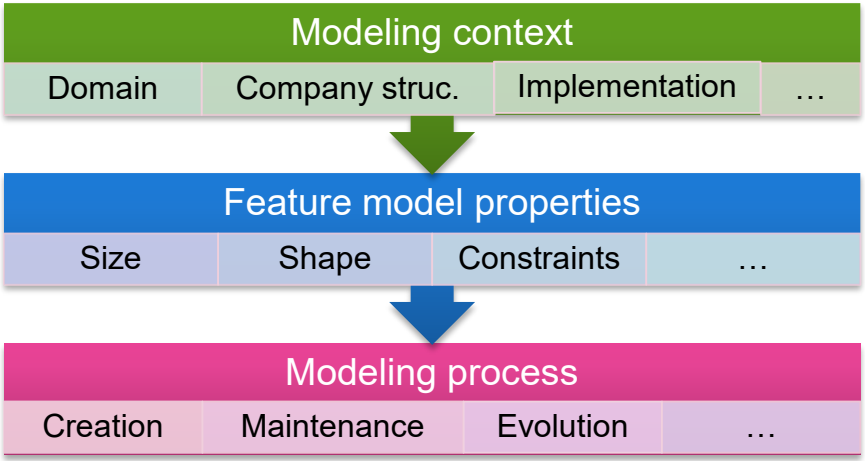
- Identified 31 paper across 26 years
- 11 different modeling notations
- 14 different application domains



Domains and types of identified papers

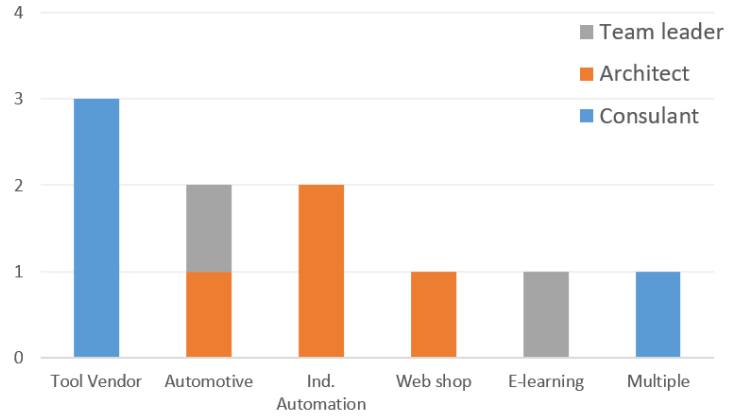
Interviews

- Semi-structured interviews (~1h)
- Included experienced practitioners, tool vendors, and consultants



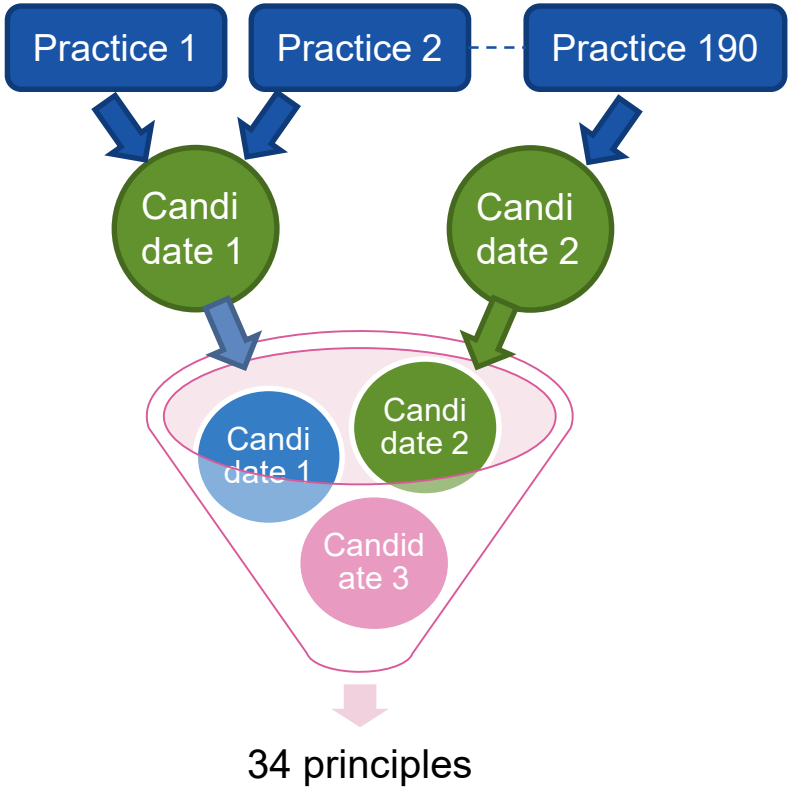
Interview guide

- 10 interviews from nine companies
- Company size 50 to 150.000 employees
- Feature model size 40 to >1000 features
- Implementation: C, C++, Java, model-based

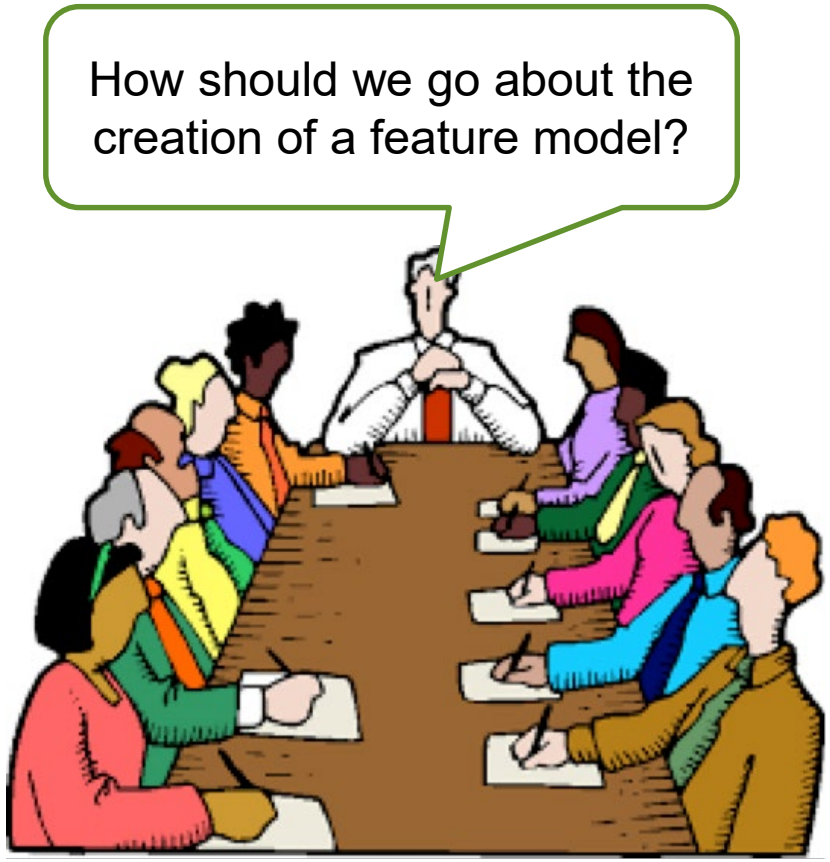


Domains and roles of interviewees

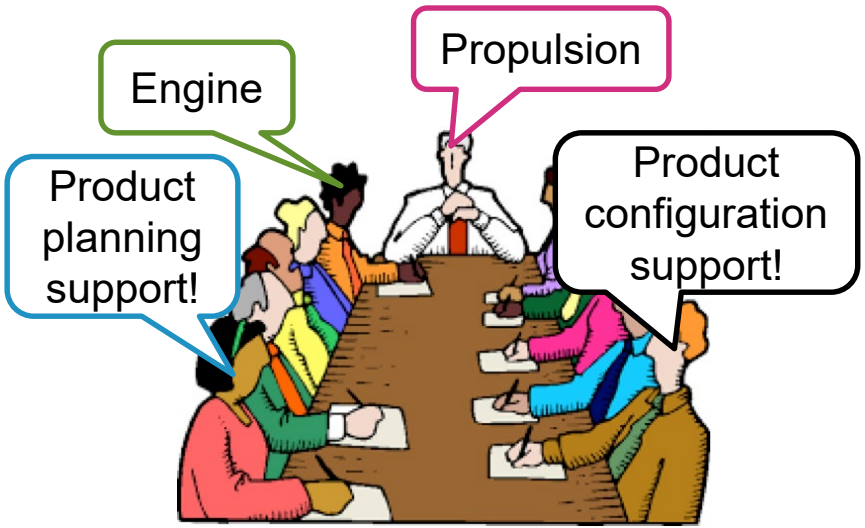
Synthesizing the principles



Category	# Principles
Planning and Preparation	6
Training	3
Information sources	1
Model Organization	5
Modeling	11
Dependencies	2
Quality Assurance	3
Model Maintenance and Evolution	3


How should we go about the creation of a feature model?

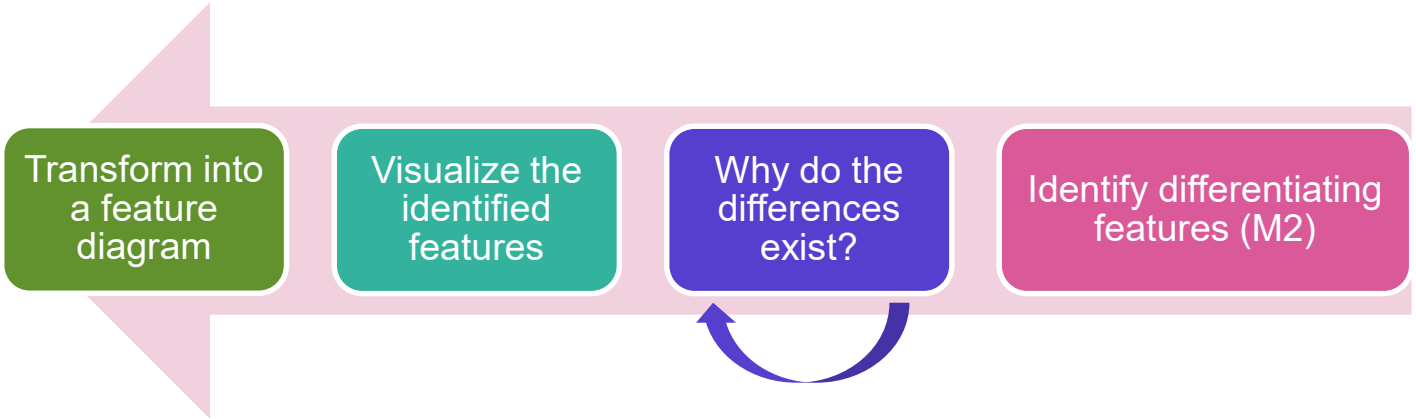
Planning and Preparation



- Identify relevant stakeholders (PP6)
  - Who are the modelers (keep the number low PP6)
  - Who are the users, maintainer
- Unify the domain terminology (PP2).
- Define the purpose of the feature model (PP3).
- Define feature to sub-feature decomposition criteria (PP4).
  - Functional, technical decomposition?

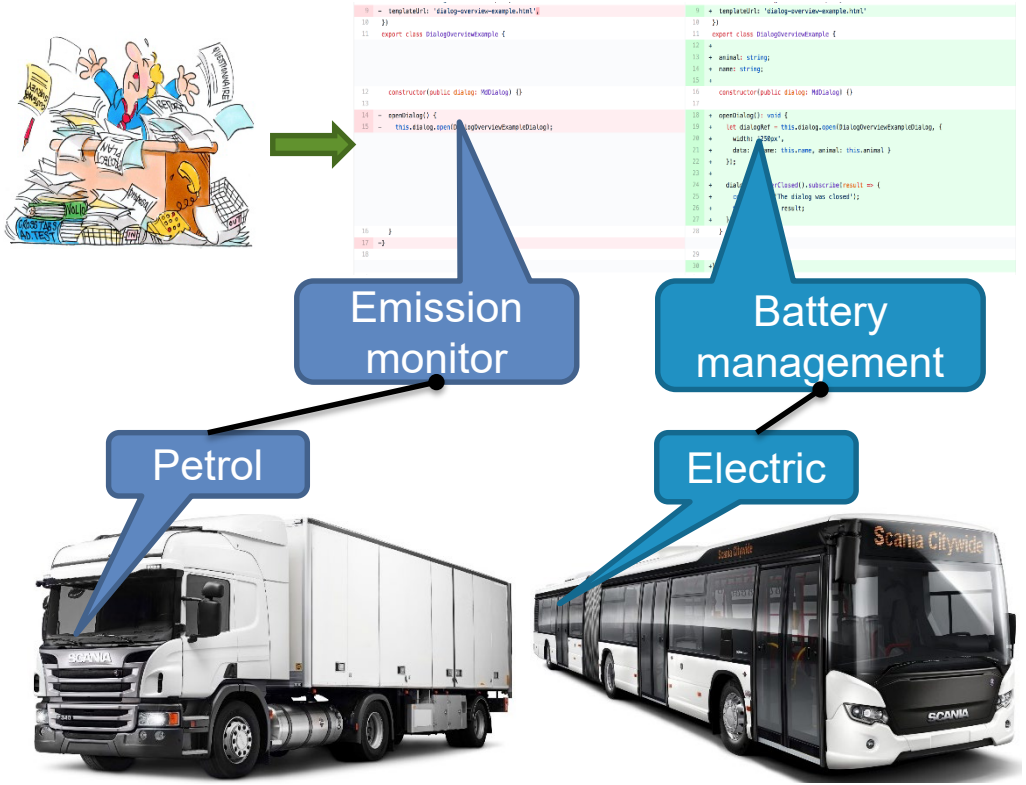
Training

- Conduct a pilot project (T3). (The “why, why, why” process)



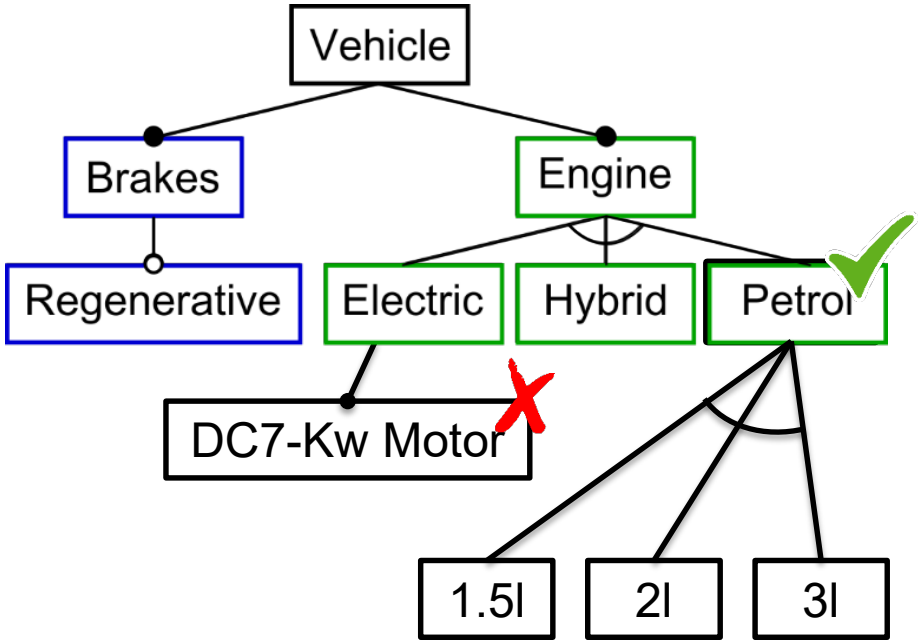
I1: ”…*keep asking why until they come up from implementation level detail which is usually where they are thinking up until the essential features.*”

Information Sources and Modeling



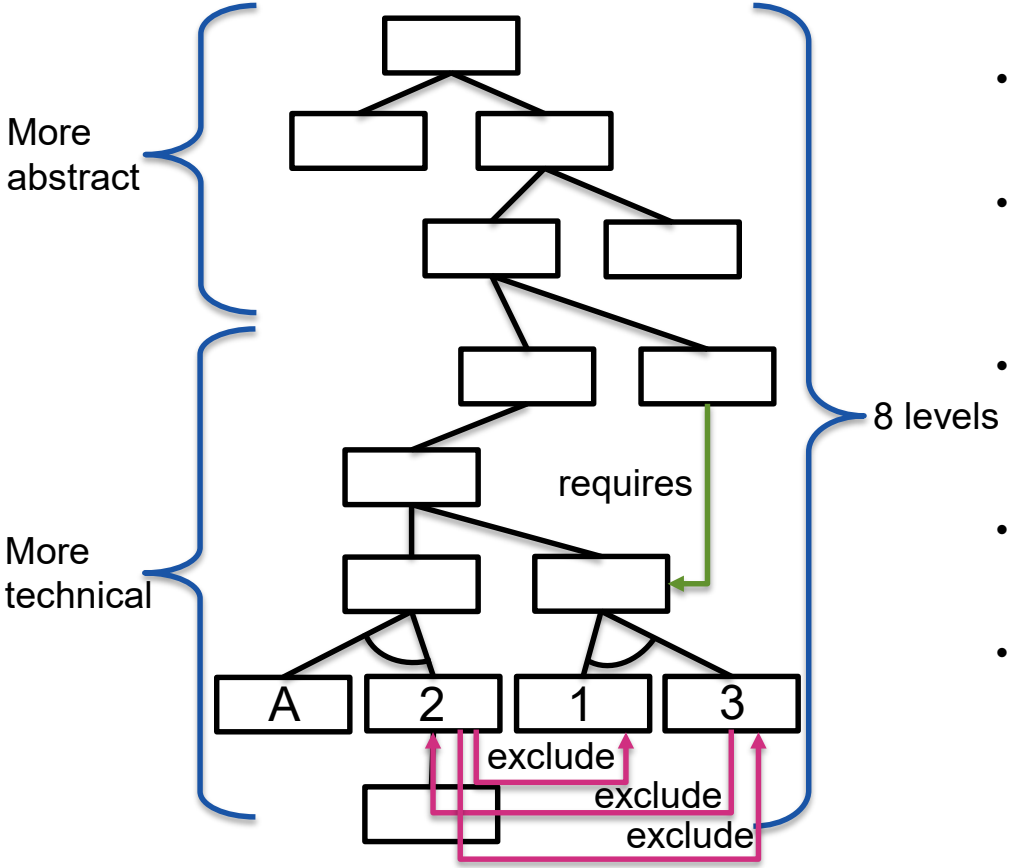
- Use domain knowledge from artifacts (IS1), in workshops with domain experts (M1).
- First identify distinguishing features (M2).
- Apply bottom-up modeling to artifacts (M3).
- Apply top-down modeling to the domain (M4).
- Apply bottom-up and top-down modeling (M5).

Modeling (continued)



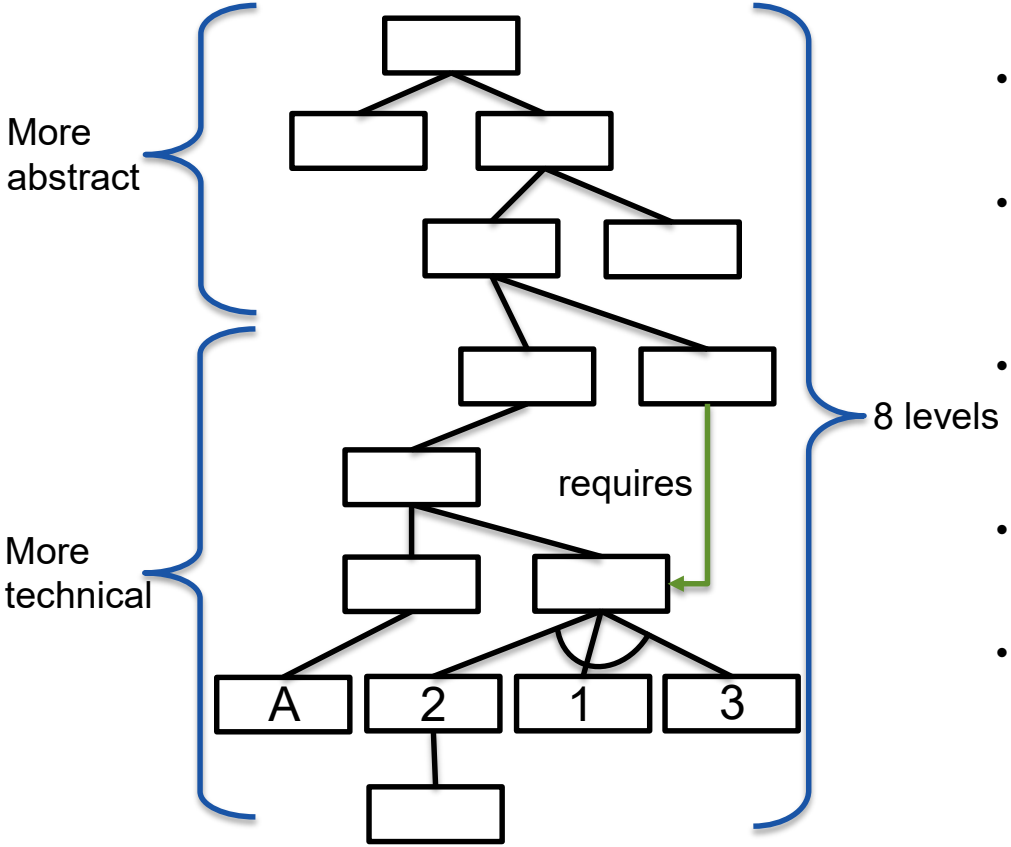
- A feature is typically a functional abstraction (M6) and Boolean features should be preferred (M10).
- Define default feature values (M8).
- Define feature-model views (M9).

Model Organization



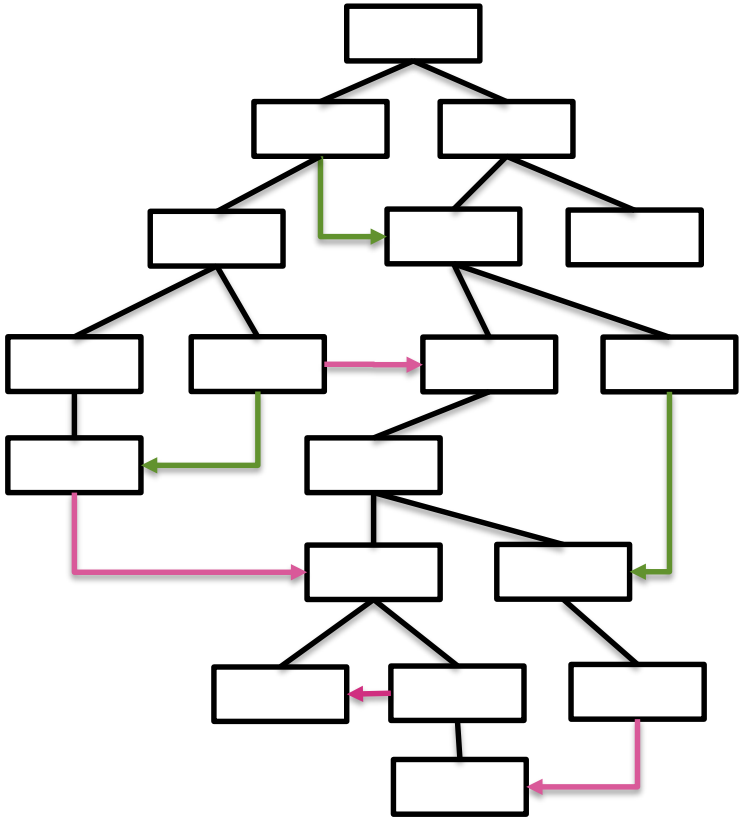
- Model hierarchy not deeper than 8 levels (MO1).
- Abstract features should be higher in the hierarchy (MO2).
- Split large models and ensure consistency with '*interface' models* (MO3).
- Avoid complex cross-tree constraints (MO4).
- Maximize cohesion and minimize coupling with feature groups (MO5).

Model Organization



- Model hierarchy not deeper than 8 levels (MO1).
- Abstract features should be higher in the hierarchy (MO2).
- Split large models and ensure consistency with '*interface' models* (MO3).
- Avoid complex cross-tree constraints (MO4).
- Maximize cohesion and minimize coupling with feature groups (MO5).

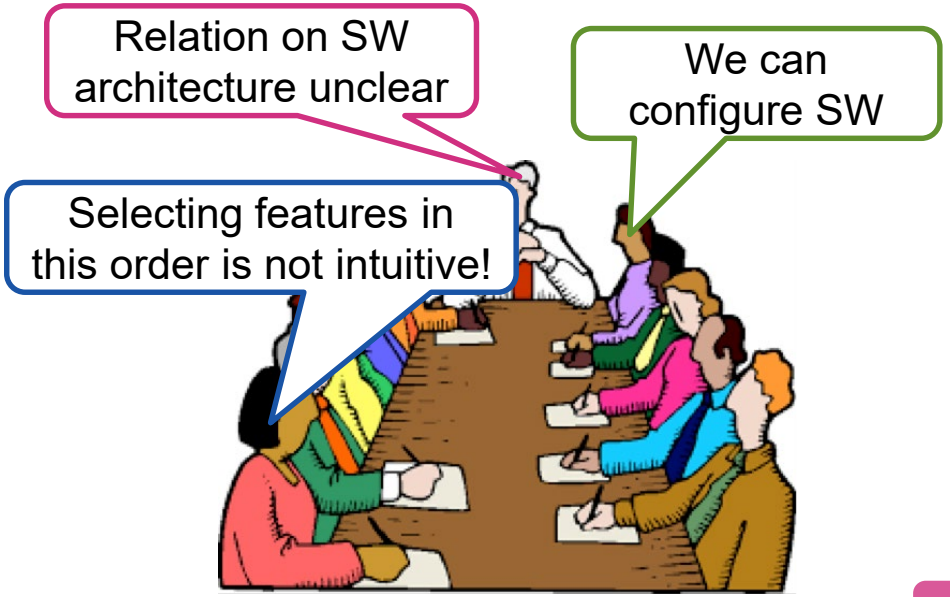
Dependencies



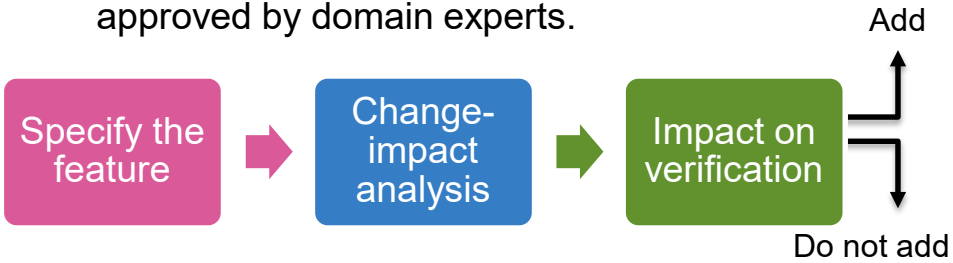
- D1: *If the models are configured by (company) experts, avoid feature-dependency modeling*
  - *~50% of features involved in a dependency*
  - *Add custom dependencies, e.g. recommends*

- D2: *If the main users of a feature model are end-users, perform feature-dependency modeling.*
  - *Support choice-propagation*
  - *Avoid invalid configurations*

Quality Assurance, Maintenance and Evolution



- Validate the feature model in workshops (QA1).
- Use the obtained feature model to derive configurations (QA2).
- Use regression tests to check for invalid configurations (QA3), if D1 applied.
- Use centralized model governance (MME1) and version manage the entire model (MME2).
- MME3: New features should be defined and approved by domain experts.

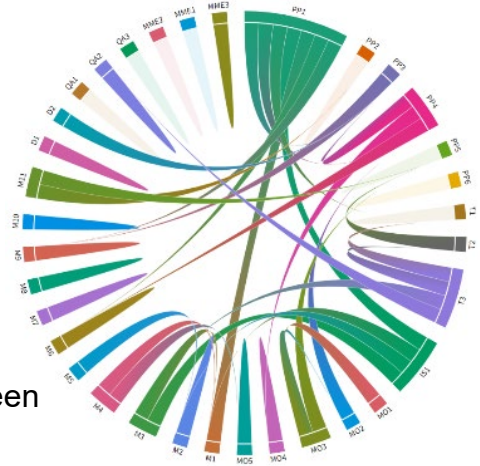


Towards a methodology for feature modeling

- The principles are ordered in the most probable sequence but we do not claim that this represents a feature-modeling process.

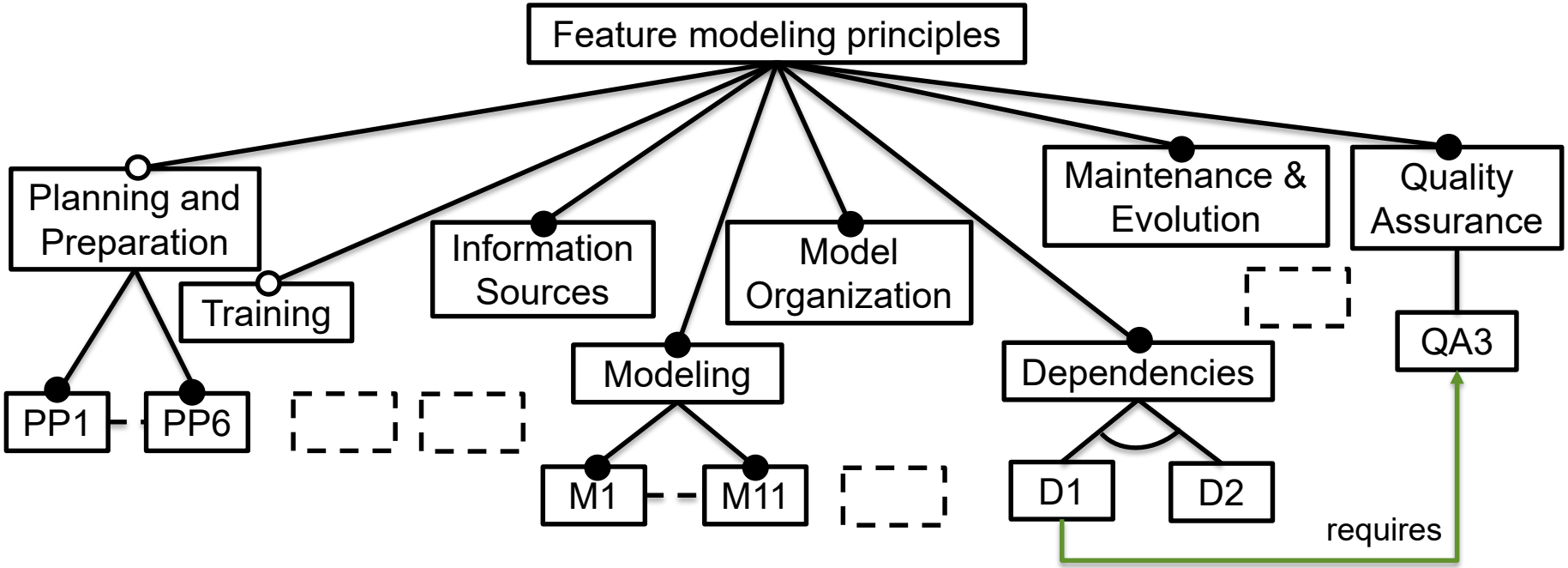
- Some principles depend on the modeling context (e.g. D1 vs D2, M9 vs MO3).

- 15 principles impact/are impacted by the application of other principles.



Dependencies between principles

Conclusion



Thank you!