



# Using Model Scoping with Expected Model Elements to Support Software Model Inspections: Results of a Controlled Experiment

Carlos Gracioli Neto<sup>1,3</sup>, Amadeu Anderlin Neto<sup>2</sup>, Marcos Kalinowski<sup>2</sup>, Daniel Cardoso Moraes de Oliveira<sup>3</sup>, Marta Sabou<sup>4</sup>, Dietmar Winkler<sup>4,5</sup>, Stefan Biffel<sup>4</sup>

<sup>1</sup> Federal Institute of Education Science and Technology of Mato Grosso, Rondonópolis, Brazil

<sup>2</sup> Pontifical Catholic University of Rio de Janeiro, Brazil

<sup>3</sup> Federal Fluminense University, Niterói, Brazil

<sup>4</sup> TU Wien, Institute of Information Systems Engineering, Vienna, Austria

<sup>5</sup> Christian Doppler Laboratory for Security and Quality Improvement in the Production System Life Cycle, Vienna, Austria.

qse.ifs.tuwien.ac.at  
www.sqi.at  
dietmar.winkler@tuwien.ac.at

# Motivation & Key Questions



## Motivation and Application Context

- § Traditional **Software Inspection** to identify defects in software engineering models.
- § Limitations for **Large-scale** software engineering models.
- § **Expected Model Elements (EMEs)** and **Model Scoping** (remove unrelated parts).

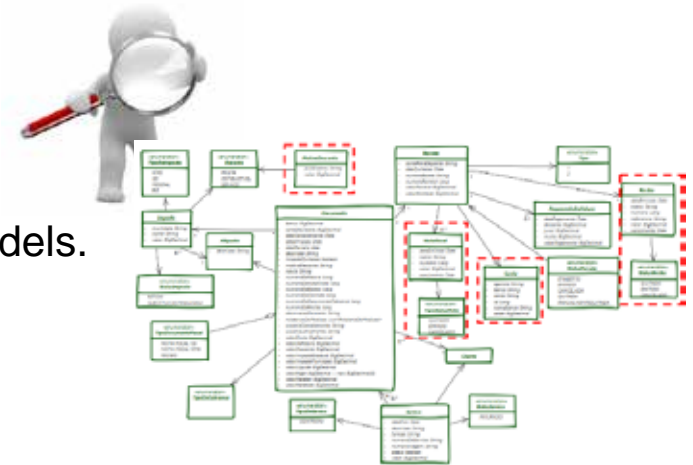


Fig. Context of Model Inspection.

## Key Question

- § How to improve defect detection performance for large-scale engineering models with model scoping?

## Goal of this presentation

- § Report on a **controlled experiment** with students **using real industrial artifacts** aiming to understand the impact of model scoping and model inspection effectiveness/efficiency.
- § Inspection of **UML class diagrams** using Model Scoping with EMEs compared to traditional Software Inspection (without model scoping and EMEs).


# Software Model Inspections

- § Software Inspection\* is a well-established formal approach for efficient defect detection in early software development phases, e.g., during software design.
- § Model Scoping is generic and not restricted to a particular type of requirements.

**1 Escopo do Módulo de Administração (MADM)**  
 O Módulo de Administração Geral integra: serviços gerais que fornecem informações básicas de cadastro (normalmente utilizadas nos demais módulos do sistema) sejam entidades (consultas, inclusão, alteração e exclusão). O software deve permitir que os clientes, de forma, ao menos de acesso, ao sistema, no âmbito de suas, de empresas, de organizações das empresas e estruturas, as informações das empresas, no âmbito, no âmbito e no caso de documentos básicos utilizados sejam tratados no sistema. Esta funcionalidade envolve a consulta, inclusão, exclusão e alteração. Deve permitir, ainda, que o personal de cadastro de MADM seja mantido no sistema. Esta funcionalidade envolve a consulta, inclusão, exclusão e alteração. Além disso, o software deve ser capaz de manter um histórico das alterações para que a informação associada ao documento (o personal vigente na época de funcionamento) não seja perdida. Destaca-se, por fim, que o módulo de administração não possui funcionalidade de integração/dependência de dados provenientes de outros módulos.

**2 Requisitos Funcionais (Reqs de Usu)**  
**RF1:** O software deve permitir que o Setor Operacional e o Setor Administrativo efetuam a manutenção de dados (consultas, inclusão, alteração e exclusão).  
**RF2:** O software deve permitir que o Setor Administrativo efetua a manutenção de dados (consultas, inclusão e alteração).  
**RF3:** O software deve permitir que o Setor Administrativo efetua a manutenção de empresas (consultas, inclusão, alteração e exclusão).  
**RF4:** O software deve permitir que o Setor Administrativo e o Setor Financeiro efetuam a manutenção de registros (consultas, inclusão, alteração e exclusão).

**2.1 Diagrama de Casos de Uso**



**2.2 Descrição de Alguns Casos de Uso**

**2.2.1 UC01 – Manter Cliente**  
**Objetivo:** Permitir que o usuário seja inserido (consulta, inclusão, alteração e exclusão) no sistema.  
**Requisitos:** SRO1  
**Atores:** Setor Operacional, Setor Administrativo  
**Trigôer:** O ator seleciona a opção Manter Cliente.  
**Fluxo Principal:**

- O sistema recebe uma tela de filtro com as seguintes informações:
  - Sigla ou
  - Cidade; - CNPJ; - Estado;
- Voltar;
- O ator preenche os filtros e seleciona a opção Buscar [BUSCAR] [A];
- O sistema recebe as telas de consulta de registros de filtro com as seguintes informações (acessar inclusão):
  - Sigla - Cidade; - CNPJ; - Endereço; - Bairro; - Município; - Estado; - CEP;
  - Inscrição Municipal; - Inscrição Estadual; - Órgão Público Federal (sigla); - Sire e Nire, default: Nire;
  - A região onde o cliente
- O sistema apresenta ao final as opções:
  - Adicionar Novo Cliente;
  - Voltar;
- O ator seleciona a opção Editar Cliente [EDITAR] [A];
- O sistema recebe uma tela de edição de dados com as seguintes informações:
  - Sigla - Cidade; - CNPJ; - Endereço; - Bairro; - Município; - Estado; - CEP;
  - Inscrição Municipal; - Inscrição Estadual; - Órgão Público Federal (sigla); - Sire e Nire, default: Nire;
  - Tabela de Pagos, contendo:

\*CNPJ: insc - CNPJ; Insc - Inscrição; \*Tipo de Pessoa (Por Nire ou por Roteador de Serviços) [RNT]

Does the model completely and correctly represent the specification?



Are there defects in the scoped model?

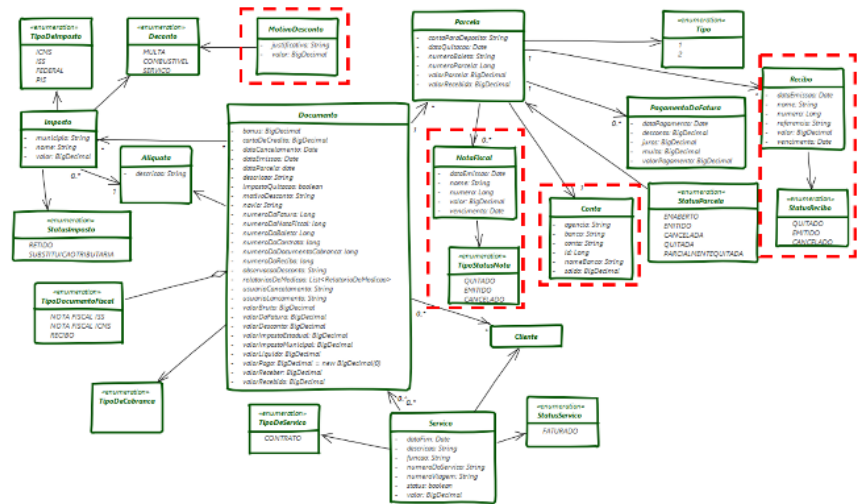


Fig. Cut-outs during Model Scoping (dashed rectangles).

Fig. Requirements Specification.

\* Fagan ME, 1976, Design and code inspection and code inspections to reduce errors in program development, IBM Systems Journal, 15(7): 182-211

# Goal & Research Question



Based on the Goal-Question Metric (GQM)\* approach:

<b>Analyze</b>	the inspection of UML class diagrams using <i>Model Scoping with EMEs</i>
<b>for the purpose of</b>	characterization
<b>with respect to</b>	inspection effectiveness & efficiency
<b>from the point of view of</b>	the information systems researcher
<b>in the context of</b>	UML class diagram inspection based on a valid functional specification, conducted by novice inspectors, when compared to not using <i>Model Scoping with EMEs</i> .



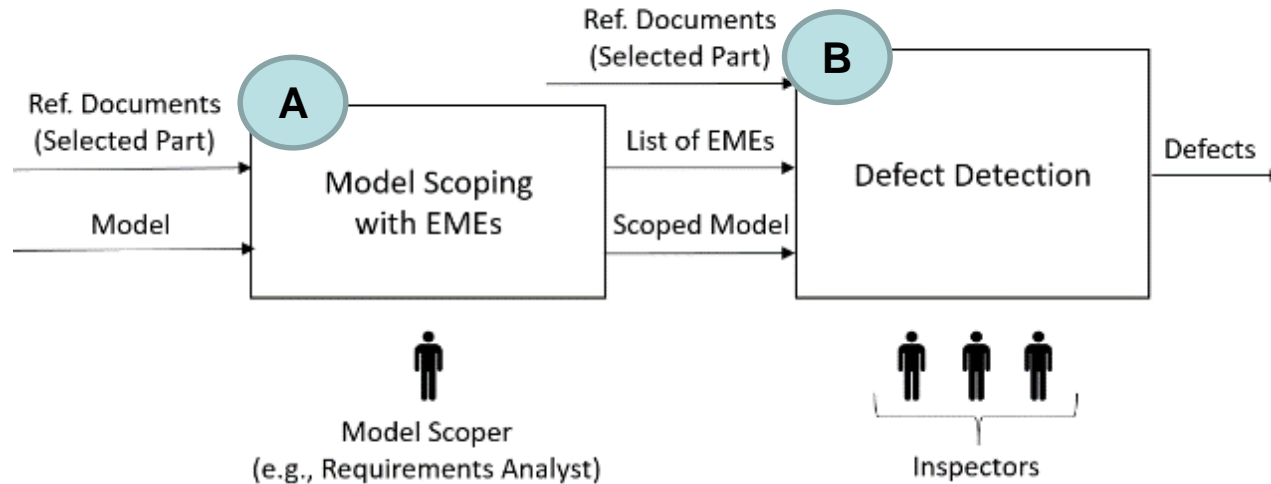
## Research Question:

- § How could a **Process for Model Scoping** based on **Expected Model Elements** as foundation for **Defect Detection** be implemented?
- § What is the **impact** of Model Scoping with EMEs on Software Inspection performance, i.e., **effectiveness and efficiency**?



\*van Solingen R, Basili V, Caldiera G, Rombach HD, 2002, Goal Question Metric (GQM) Approach, In: *Encyclopedia of Software Engineering*

# Model Scoping and Defect Detection Process with EMEs



## Step A: Model Scoping with EMEs approach includes:

1. Define the **types of EMEs**, e.g., for UML Class Diagrams: classes, attributes, relations.
2. Identify **list of relevant EMEs** based on the selected part of the reference document.
3. **Scope the model** by removing model elements that are not in the list/scope of EMEs.

## Step B: Defect Detection based on EMEs and the Scoped Reference Document:

1. **Appearance**: Is the EME represented in the model?
2. **Correctness**: Is the EME modeled correctly?

à Foundation for Identifying and Reporting Defects.

# Empirical Study

## Approach and Study Design



§ **Controlled Experiment in class-room settings.**

§ **Comparison of Defect Detection Approaches**

- Traditional (ad-hoc) inspection approach without any specific reading technique.
- With / Without Model Scopes based on EMEs.

	Exercise A - step one -	Exercise B - step two -
Group 1	Ad-hoc	Model Scoping
Group 2	Model Scoping	Ad-hoc

§ **Cross-Over Design with 2 Groups and 2 Exercises.**

- Group 1: Ad-hoc **without** Model Scopes à Ad-hoc **with** Model Scopes.
- Group 2: Ad-hoc **with** Model Scopes à Ad-hoc **without** Model Scopes.

§ **Study Schedule (3 days)**

- Day 1: Preparation - Consent Form and Characterization Questionnaire.
- Day 2: Training and 1<sup>st</sup> part of the experiment.
  - Tutorial & Training (15 min).
  - Execution of **Exercise A with focus on four simple use cases** (75 min).
- Day 3: Execution of Exercise B with **focus on two complex use cases** (75 min).

# Study Design

## Application Domain and Material



### § Application Domain

- Integrated administration system with 2 modules including
  - Simple Administrative Tasks (4 Tasks, Exercise A)  
e.g., maintaining company and customer data, tax information, and cost centers.
  - Complex Billing Tasks (2 Tasks, Exercise B)  
e.g., registering invoices for provided services; registering payments for invoices.

### § Inspection Artifacts

- Overview description; List of functional requirements; use case diagrams; and use case descriptions.
- Class diagram: 19 classes (full UML diagram) vs. 12 classes (scoped UML diagram) for the selected model scope.

### § Questionnaires

- Consent form and participant characterization (participant background).
- Qualitative Feedback following the Technology Acceptance Method (TAM)\*.

\*Turner M, Kitchenham B, Brereton P, 2010, Does the technology acceptance model predict actual use? A systematic literature review, Information and Software Technology, vol, 52: 463-479

# Study Design

## Defects and Participants



§ Requirements Specification was considered to be correct.

### § Seeded Defects in the Class Diagram

- Overall 28 seeded defects.
- Different defect types: ambiguity, incorrect facts, omission, extraneous information, (and inconsistencies\*).
- 7 typical defects per defect type at different severity levels.

Fig. Example: Qualification of 2nd experiment run.

### § Participants

- Overall 40 Participants in two experiment runs (32 + 8).
- Exact replication in the 2<sup>nd</sup> run.
- Randomized and balanced assignment to experiment groups.
- Background characterization to capture experience on (a) Software Development, (b) UML Modeling, and (c) Software Inspection.

Group	ID	Software Development	UML Models	Software Inspection
1	P33	H	H	L
	P34	L	H	L
	P35	M	M	L
	P36	H	H	M
2	P37	H	H	M
	P38	H	H	L
	P39	L	H	L
	P40	M	H	L

\* No seeded defects for inconsistencies



# Study Design

## Variables and Hypothesis



### Variables

#### § Independent Variables:

- Defect Detection approach applied, participant qualification.

#### § Dependent Variables:

- **Effectiveness**: Share of identified (true) defects and seeded defects.
- **Efficiency**: Real defects per time interval (e.g., per hour).

### Hypothesis:

§ H01: No difference regarding **defect detection effectiveness** when inspecting UML class diagrams with or without using *Model Scoping with EMEs*.

§ H02: No difference regarding **defect detection efficiency** when inspecting UML class diagrams with or without using *Model Scoping with EMEs*.

### Statistical Evaluation

§ Descriptive Statistics, Hypothesis testing based on Mann-Whitney Test at 90%\*.

\*Dybå T, Kampenes VB, Sjøberg DIK, 2006, A systematic review of statistical power in Software Engineering experiments, Information and Software Technology 48 (8):745-755.

# Results

## Defect Detection Effectiveness



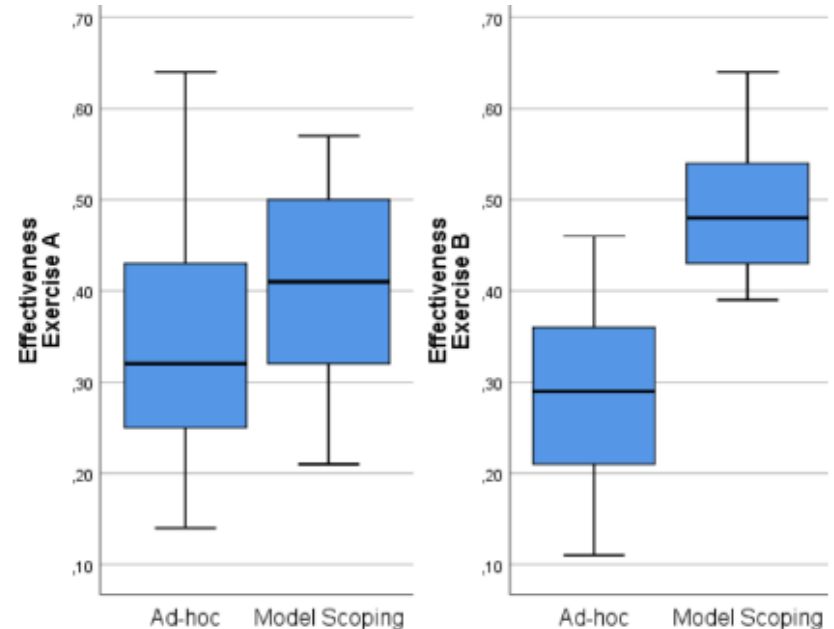
§ **Effectiveness:**

- True Defects Found / Seeded Defects.

§ **Hypothesis Testing**

- Mann-Whitney-Test (90%).
- p-value: 0.075 (s) for Exercise A.
- P-value: 0.001 (s) for Exercise B.

§ Model Scoping Groups with EME guidance were **significant more effective in both trials** (exercise A and B).



Effectiveness	Exercise A (Simple)		Exercise B (Complex)	
	Ad-Hoc	Model-Scoping	Ad-Hoc	Model-Scoping
MEAN	0,3	0,4	0,3	0,5
SD	0,13	0,10	0,10	0,07

§ **Higher Effectiveness for Defect Detection for Model Scoping Groups**  
 à **H01 must be rejected.**

# Results

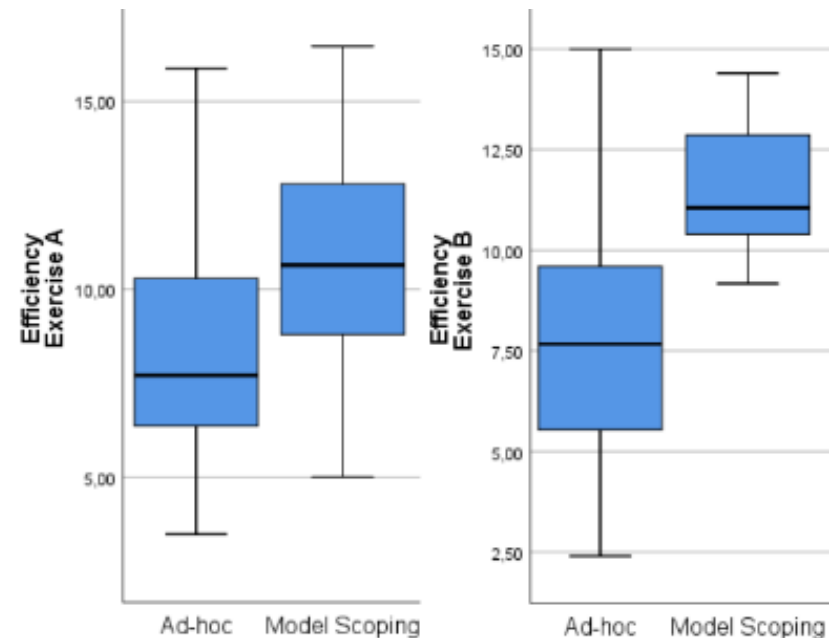
## Defect Detection Efficiency



- § **Efficiency:**
- Number of identified true defects per time interval (i.e., per hour).

- § **Hypothesis Testing**
- Mann-Whitney-Test (90%)
  - p-value: 0.025 (s) for Exercise A
  - p-value: 0.001 (s) for Exercise B

- § Model Scoping Groups with EMEs guidance were **significant more efficient in both trials** (exercise A and B).



Efficiency	Exercise A (Simple)		Exercise B (Complex)	
	Ad-Hoc	Model-Scoping	Ad-Hoc	Model-Scoping
MEAN	8,6	10,6	7,4	11,5
SD	3,15	2,80	3,10	1,73

- § **Higher Defect Detection Efficiency for Model Scoping Groups à H02 must be rejected.**

# Discussion & Threats to Validity



## How to improve defect detection performance for large-scale engineering models with model scoping?

- § In the study context the **Model Scoping and Defect Detection Process**
  - supported defect detection performance well, i.e., significantly improved defect detection effectiveness and efficiency.
- § Model Scopes and guidance by EMEs (based on feedback questionnaire):
  - was **perceived useful** by participants.
  - **decreased task complexity** based subjective participant assessment.
  - Guidance with EMEs also **supports defect detection**.

## Threats to validity

- § **Internal**: individual inspection of participants (no communication allowed), review of the experimental material and pilot test runs of the experiment.
- § **External**: focus on real-world artifacts (from an individual organization); students act as participants (we captured their experience prior to the study).
- § **Construct**: we applied a cross-over design to isolate learning effects; defects were seeded according to experiences of researchers and practitioners.
- § **Conclusion**: We removed outliers and applied statistical tests, proven in similar contexts.

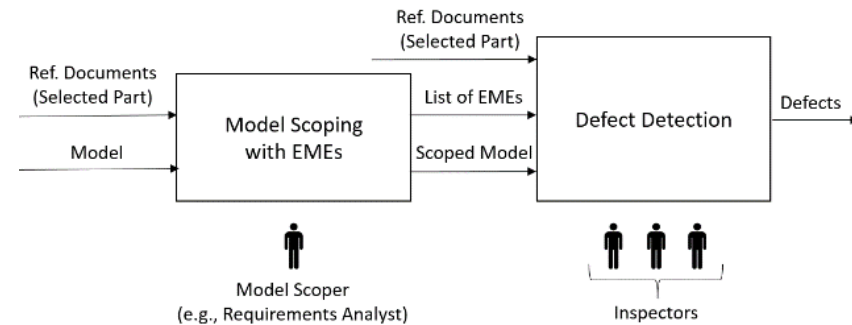
# Summary and Future Work



## Summary

§ The **Model Scoping and Defect Detection Process with EMEs** consist of a

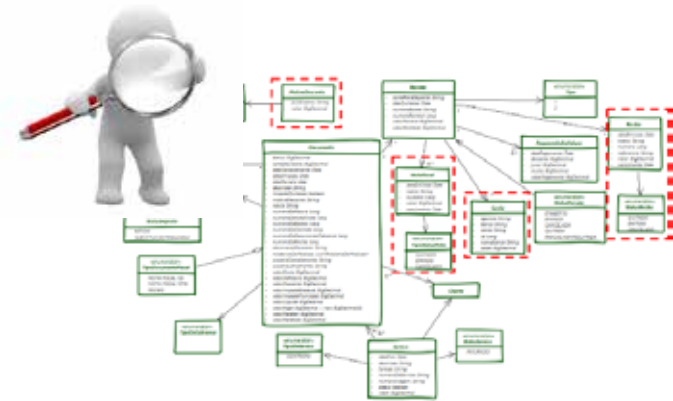
- **Model Scoping** and
- **Defect Detection Process Step.**



§ **Model Scoping** can act as filter or view to focus on relevant model elements.

§ Support for inspecting **Large-Scale Engineering Models.**

§ Promising results in the study context.



## Future Work

§ Further investigations to precisely **estimate in which cases** Model Scoping with EMEs **would be (most) worthwhile** the upfront investment.

§ **Replicating the reported experiment** on Model Scoping with EMEs, **including other engineering model types in different contexts**, to reinforce experimental evidence and improve external validity.

Thank you ...



## Using Model Scoping with Expected Model Elements to Support Software Model Inspections: Results of a Controlled Experiment

Carlos Gracioli Neto<sup>1,3</sup>, Amadeu Anderlin Neto<sup>2</sup>, Marcos Kalinowski<sup>2</sup>, Daniel Cardoso Moraes de Oliveira<sup>3</sup>, Marta Sabou<sup>4</sup>, Dietmar Winkler<sup>4,5</sup>, Stefan Biffel<sup>4</sup>

<sup>1</sup> Federal Institute of Education Science and Technology of Mato Grosso, Rondonópolis, Brazil

<sup>2</sup> Pontifical Catholic University of Rio de Janeiro, Brazil

<sup>3</sup> Federal Fluminense University, Niterói, Brazil

<sup>4</sup> TU Wien, Institute of Information Systems Engineering, Vienna, Austria

<sup>5</sup> Christian Doppler Laboratory for Security and Quality Improvement in the Production System Life Cycle, Vienna, Austria.

qse.ifs.tuwien.ac.at

www.sqi.at

dietmar.winkler@tuwien.ac.at