Towards an Experiment Line on Software Inspection with Human Computation

Stefan Biffl[1], Marcos Kalinowski[2], Dietmar Winkler[1,3]

[1] TU Wien, Institute of Information Systems Engineering, Vienna, Austria

[2] Pontifical Catholic University of Rio de Janeiro, Brazil

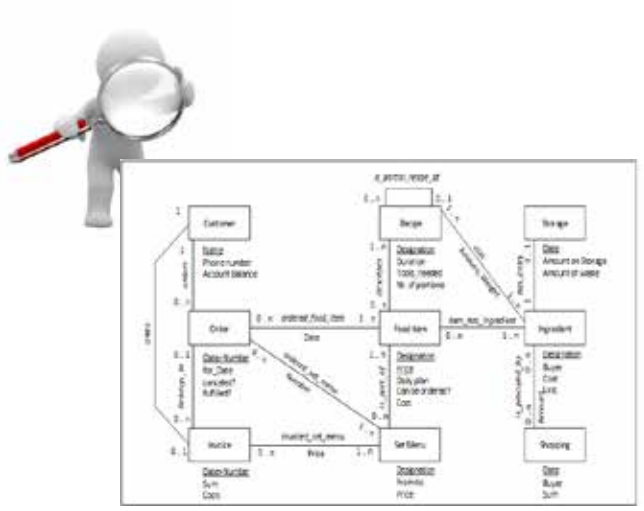
[3] Christian Doppler Laboratory for Security and Quality Improvement in the Production System Life Cycle, Vienna, Austria.

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 design documents and models.
- Limited resources for inspection processes.
- Large-scale software engineering models.



Model Inspection

Key questions

- How to handle large-scale engineering models with limited resources in a human computation context?
- How could an experiment line address variability for planning experiments?
- What are the expected benefits of an experiment line?

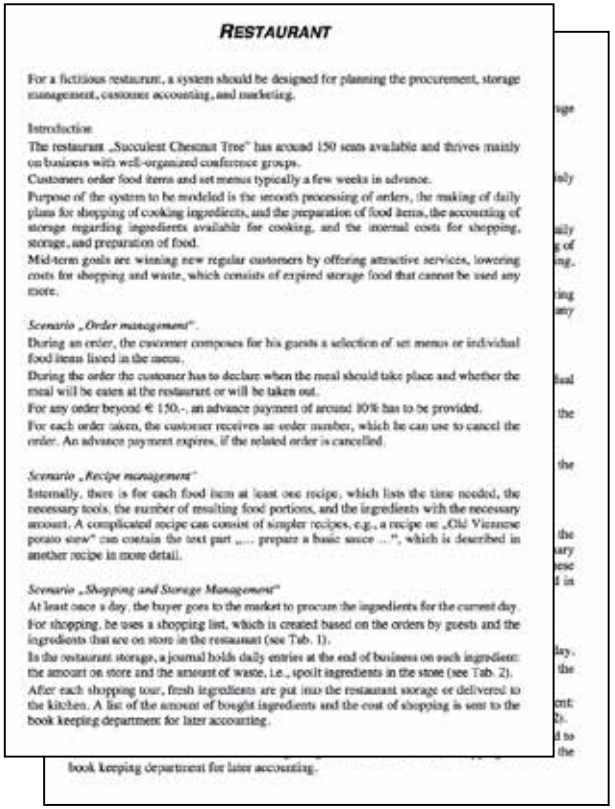
Goal of this presentation

- Software Inspection Process with Human Computation (Crowdsourcing Supported Inspection (CSI)).
- Managing variability of a family of experiments in academic / industry environment.

Illustrative Example .. Starting point

Inspection Task

- Input: Reference document, e.g., requirements specifications.
- Task: Identify defects in (large-scale) models early, effective, and efficient.
- Output: True defects in the model.



RESTAURANT

For a fictitious restaurant, a system should be designed for planning the procurement, storage management, customer accounting, and marketing.

Introduction

The restaurant „Succulent Chestnut Tree" has around 150 seats available and thrives mainly on business with well-organized conference groups.
Customers order food items and set menus typically a few weeks in advance.
Purpose of the system to be modeled is the smooth processing of orders, the making of daily plans for shopping of cooking ingredients, and the preparation of food items, the accounting of storage regarding ingredients available for cooking, and the internal costs for shopping, storage, and preparation of food.
Mid-term goals are winning new regular customers by offering attractive services, lowering costs for shopping and waste, which consists of expired storage food that cannot be used any more.

Scenario „Order management".
During an order, the customer composes for his guests a selection of set menus or individual food items listed in the menu.
During the order the customer has to declare when the meal should take place and whether the meal will be eaten at the restaurant or will be taken out.
For any order beyond € 150,-, an advance payment of around 10% has to be provided.
For each order taken, the customer receives an order number, which he can use to cancel the order. An advance payment expires, if the related order is cancelled.

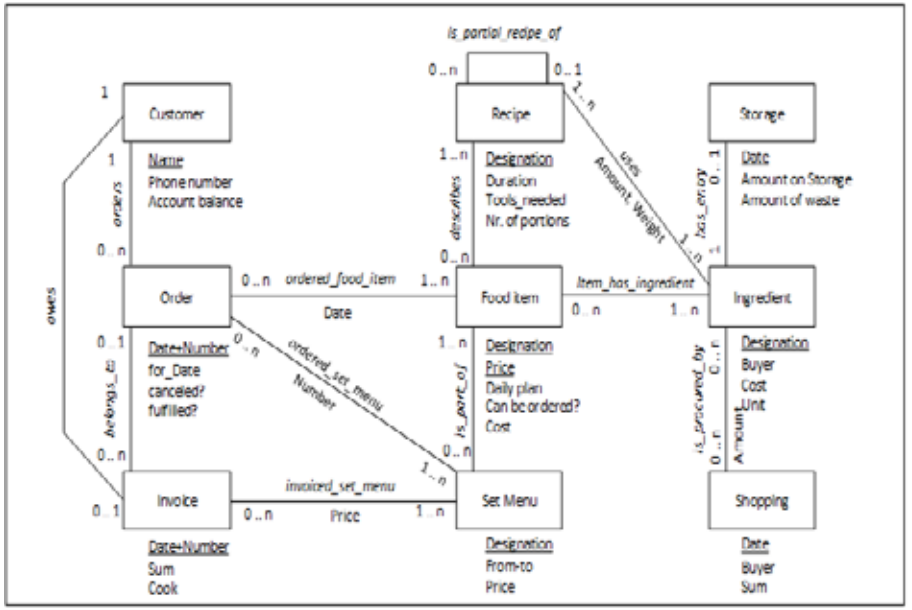
Scenario „Recipe management"
Internally, there is for each food item at least one recipe, which lists the time needed, the necessary tools, the number of resulting food portions, and the ingredients with the necessary amount. A complicated recipe can consist of simpler recipes, e.g., a recipe on „Old Viennese potato stew" can contain the text part „... prepare a basic sauce ...", which is described in another recipe in more detail.

Scenario „Shopping and Storage Management"
At least once a day, the buyer goes to the market to procure the ingredients for the current day.
For shopping, he uses a shopping list, which is created based on the orders by guests and the ingredients that are on store in the restaurant (see Tab. 1).
In the restaurant storage, a journal holds daily entries at the end of business on each ingredient: the amount on store and the amount of waste, i.e., spoilt ingredients in the store (see Tab. 2).
After each shopping tour, fresh ingredients are put into the restaurant storage or delivered to the kitchen. A list of the amount of bought ingredients and the cost of shopping is sent to the book keeping department for later accounting.

Requirements Specification

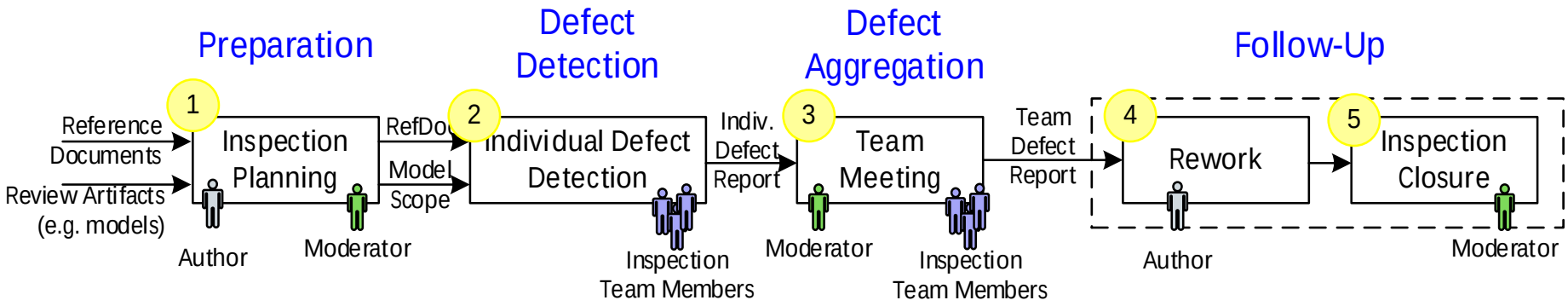
Does the model completely and correctly represent the specification?

Are there defects in the model?



System EER Diagram Model

Traditional (Best-Practice) Inspection



Benefits:

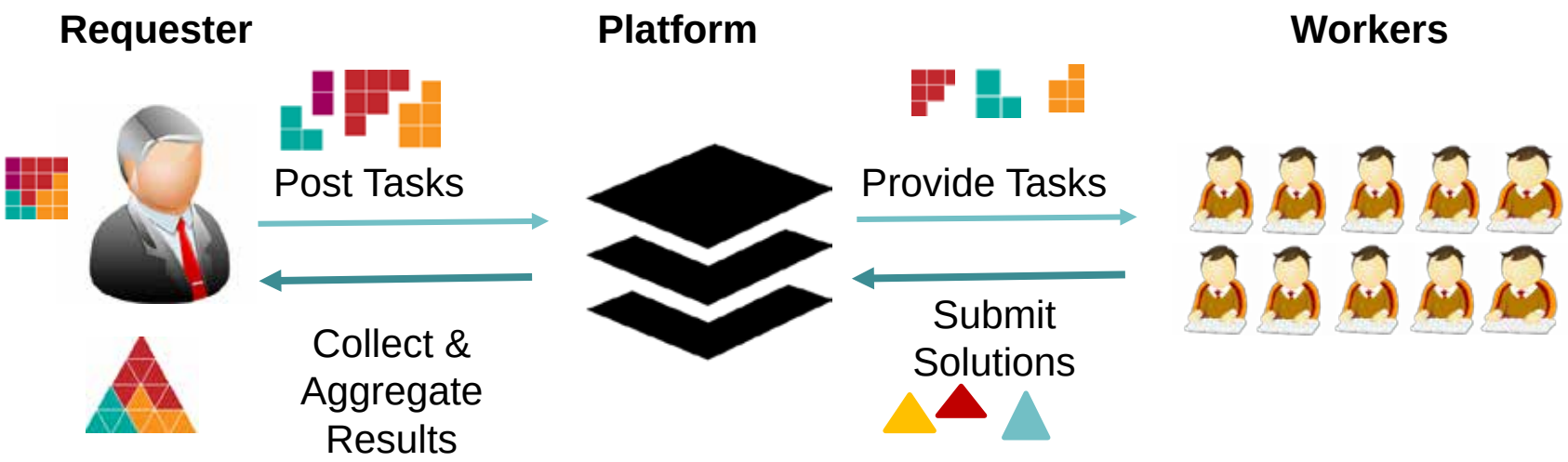
- Formal and structured process approach (five inspection phases) to identify defect early and efficient in engineering artifacts.
- Well established and investigated process approach.
- Guidelines and reading techniques support defect detection, e.g., perspectives or scenarios.

Limitations:

- Typically (expensive) experts are part of the inspection team.
- Limited resources (e.g., 2h of inspection) à for inspecting large-scale documents need for several inspection cycles and coordination.
- Limited tool support.

Human Computation & Crowdsourcing

"The act of undertaking any external software engineering tasks by an undefined, potentially large group of online workers in an open call format." (Mao et al., 2016)



- Crowdsourcing (CS) mechanism has been applied in software engineering planning and analysis, implementation, maintenance, and testing ..
- .. but very limited in the area of Software Quality Assurance and Software Inspection.

K. Mao, L. Capra, M. Harman, Y. Jia. A survey of the use of crowdsourcing in software engineering. Journal of Systems and Software, 2016.

Research Questions

Objectives and Approach

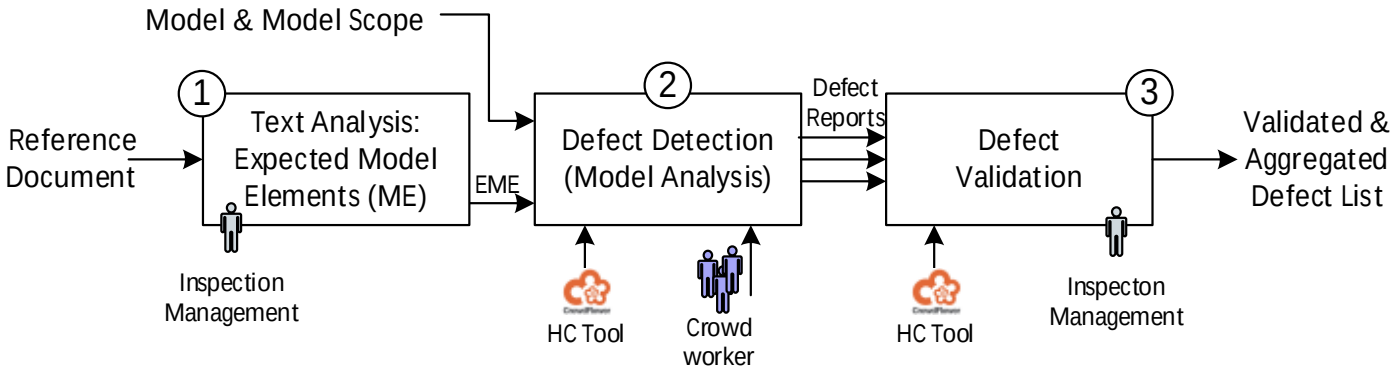
- Support of software inspection tasks with crowdsourcing techniques in context of a family of experiment.
- Key Elements:
  - Splitting up inspection tasks into small pieces of work,
  - Distributing inspection work load to a crowd of workers and/or experts within an organization,
  - Providing tool support.
  - Variability model for (inspection) experiment planning.

Research Questions

- RQ.1: How to handle large-scale engineering models with limited resources in a human computation context?
- RQ.2: How could an experiment line address variability for planning experiments?
- RQ.3: What are the expected benefits for such an experiment line?

Software Inspection with Human Computation

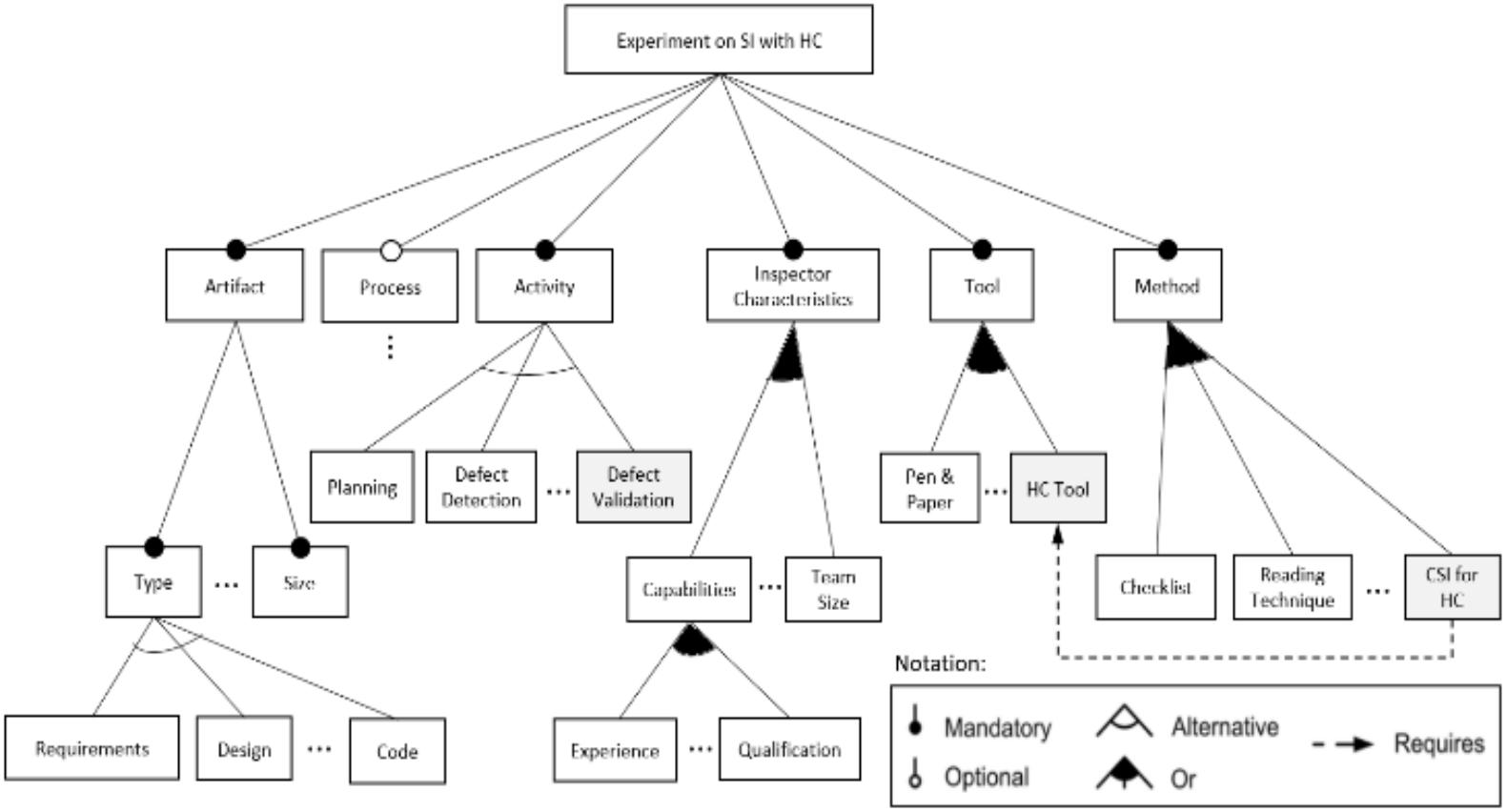
RQ.1: How to handle large-scale engineering models with limited resources in a human computation context?



1. **Text Analysis:**
   - § Identification of Model Elements (i.e., Entities, Relationships, Attributes) that represent the building blocks of a model à foundation for defect detection.
2. **Model Analysis (Defect Detection):**
   - § Based on ME, defect detection in the model under inspection à candidate defect reports by individual crowd workers.
3. **Defect Validation ("Team Meeting"):**
   - § Validation of reported defects derived from previous model analysis tasks. à justified and validated defects by crowd workers.

Feature Model for Software Inspection with Human Computation

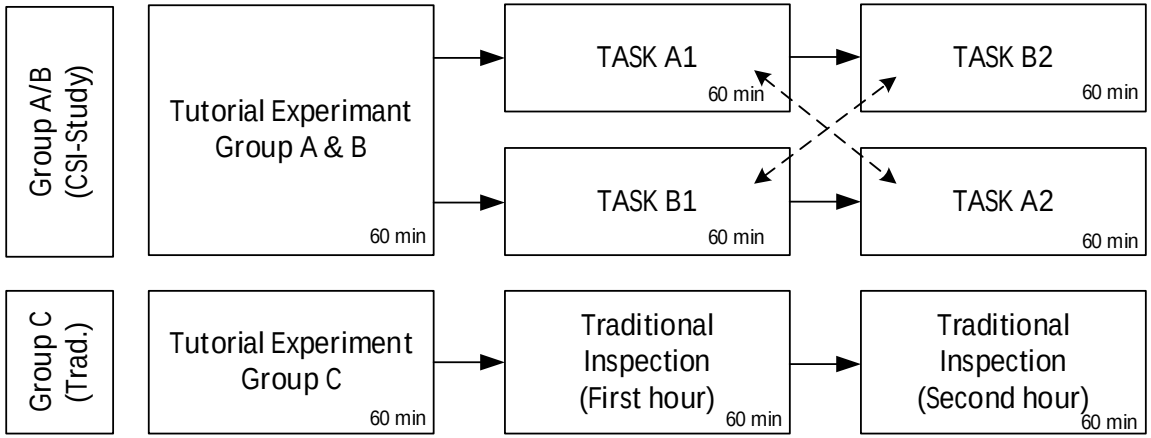
RQ.2: How could an experiment line address variability for planning experiments?



 Based on Software Inspection Theory.

Study Setup and Design

Basic Study Design



- Study Type: Controlled Experiment
- CSI process vs. Traditional Best-Practice Inspection (control group) with cross-over design.
- >75 participants in academic course in 4 different experiment runs
- Study Material:
  - Design Specification: 3 pages, 7 scenarios and 110 MEs.
  - EER Diagram: 9 entities, 13 relationships, 32 attributes; 33 seeded defects.
  - Questionnaires (experience and feedback), guidelines for task execution.
  - Tool: *Crowdflower/Figure Eight* [1] application and configuration.

[1] Crowdflower: www.crowdflower.com / Figure Eight: www.figure-eight.com/

Study History

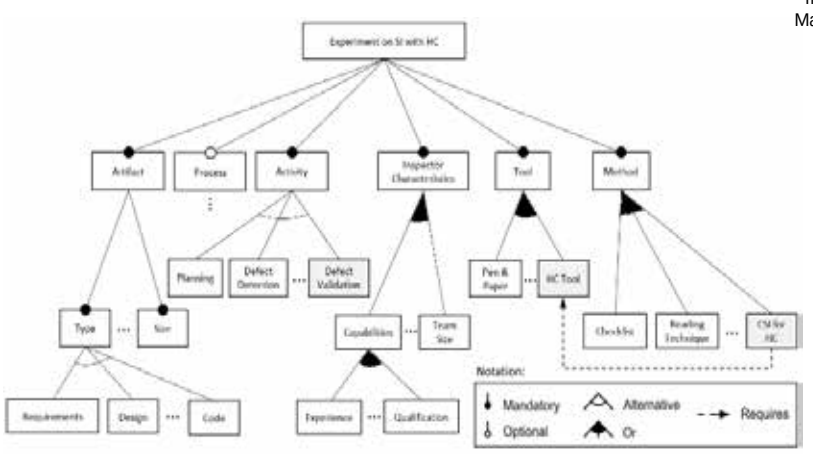
- Four different experiment runs until today.
  - Fall 2016, spring 2017, fall 2017, spring 2018 (currently running).
- Similar / slightly improved material: Reference Document (Scenarios), Inspection Artifact (EER Model), Experience Questionnaire, Feedback Questionnaires, Artefact Scale, Seeded Defects, Process for traditional (pen & paper inspection)
- Implemented Variations:
  - Focus on CSI process improvement, scope, and tooling improvements

Experiment Runs:	Fall 2016	Spring 2017	Fall 2017	Spring 2018
Process Steps:	Task and Scope Cross Over		Scope Cross Over Design	
+ Text Analysis	TASK Ax	TASK Ax	-	-
+ Model Analysis	TASK Bx	TASK Bx	TASK A1	TASK B2
+ Defect Validation	-	-	TASK B1	TASK A2
Tooling:			Improved Task Assignment and Data Collection for CSI Implementation	
+ Text Analysis	free-text data collection (Crowdflower)	free-text data collection (Crowdflower)		
+ Model Analysis	free-text data collection (Crowdflower)	free-text data collection (Crowdflower)	guided data collection (Crowdflower)	guided data collection (Figure Eight)
+ Defect Validation	-	-	guided data collection (Crowdflower)	guided data collection (Figure Eight)

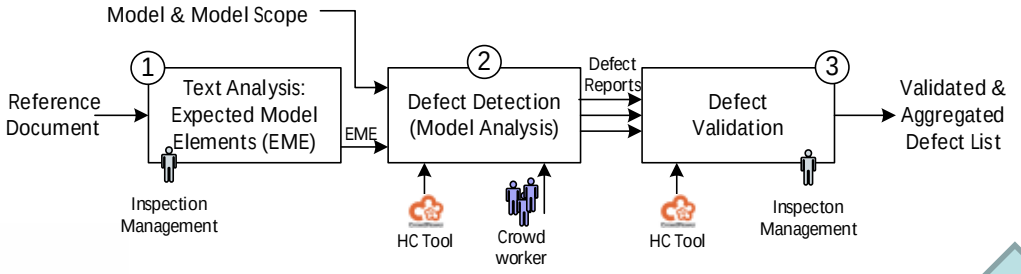
Expected Benefits

RQ.3: What are the expected benefits for such an experiment line?

- Support of strategically planning of a family of studies to facilitate the cooperation of research groups.
- Supporting systematic reuse of experiment design and material.
- Foundation for an aggregation strategy of experiment results (towards a Body of Knowledge).
- Integration of industry studies as integral part of the family of experiments based on a proven experiment setup.



Feature Model for Strategic Planning of Study Variants



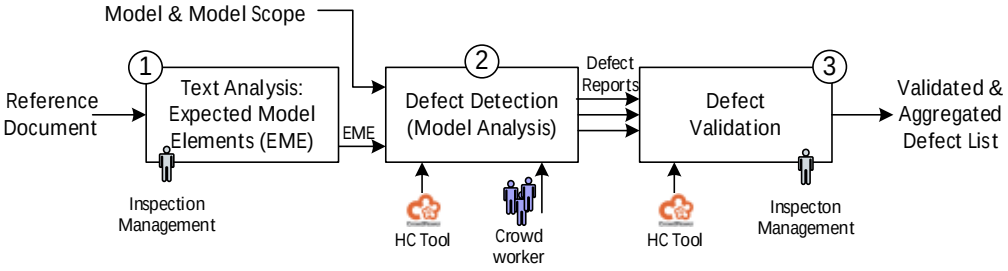
Study Context: Adapted Inspection Process

Specific Study Instance and Results

Summary and Future Work

Summary

- Improving traditional Software Inspection with Human Computation, i.e., Crowdsourcing Based Inspection (CSI).
- Feature Models support planning empirical studies in academia and industry based on a stable study architecture.



Future Work

- Along the planning of a family of empirical studies we will focus on:
  - Needs from candidate industry partners to improve defect detection within a defined context (Case Studies?).
  - Various model types, e.g., structural and behavior models in different domains.
  - Various model sizes towards large-scale software engineering models.

Thank you ...

Towards an Experiment Line on Software Inspection with Human Computation

Stefan Biffl[1], Marcos Kalinowski[2], Dietmar Winkler[1,3]

[1] TU Wien, Institute of Information Systems Engineering, Vienna, Austria
[2] Pontifical Catholic University of Rio de Janeiro, Brazil
[3] Christian Doppler Laboratory for Security and Quality Improvement in the Production System Life Cycle, Vienna, Austria.

qse.ifs.tuwien.ac.at/~winkler
www.sqi.at
Dietmar.Winkler@tuwien.ac.at