# Agile Approach and MDA in Software Development Process

Jaroslava Kniežová, Assoc. Prof, Ing, PhD.

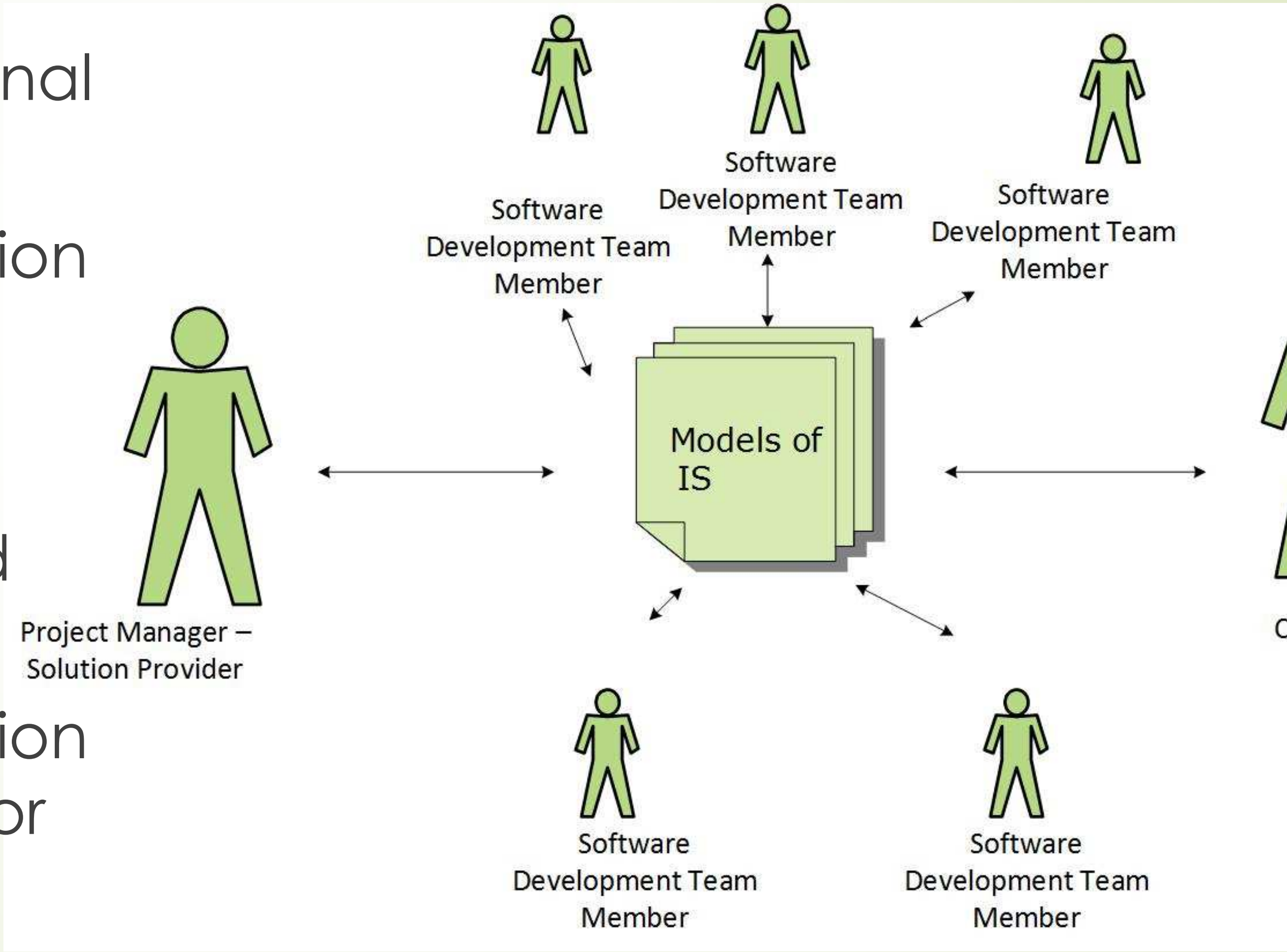
Comenius University, Faculty of Management, Bratislava Slovakia

# Overview

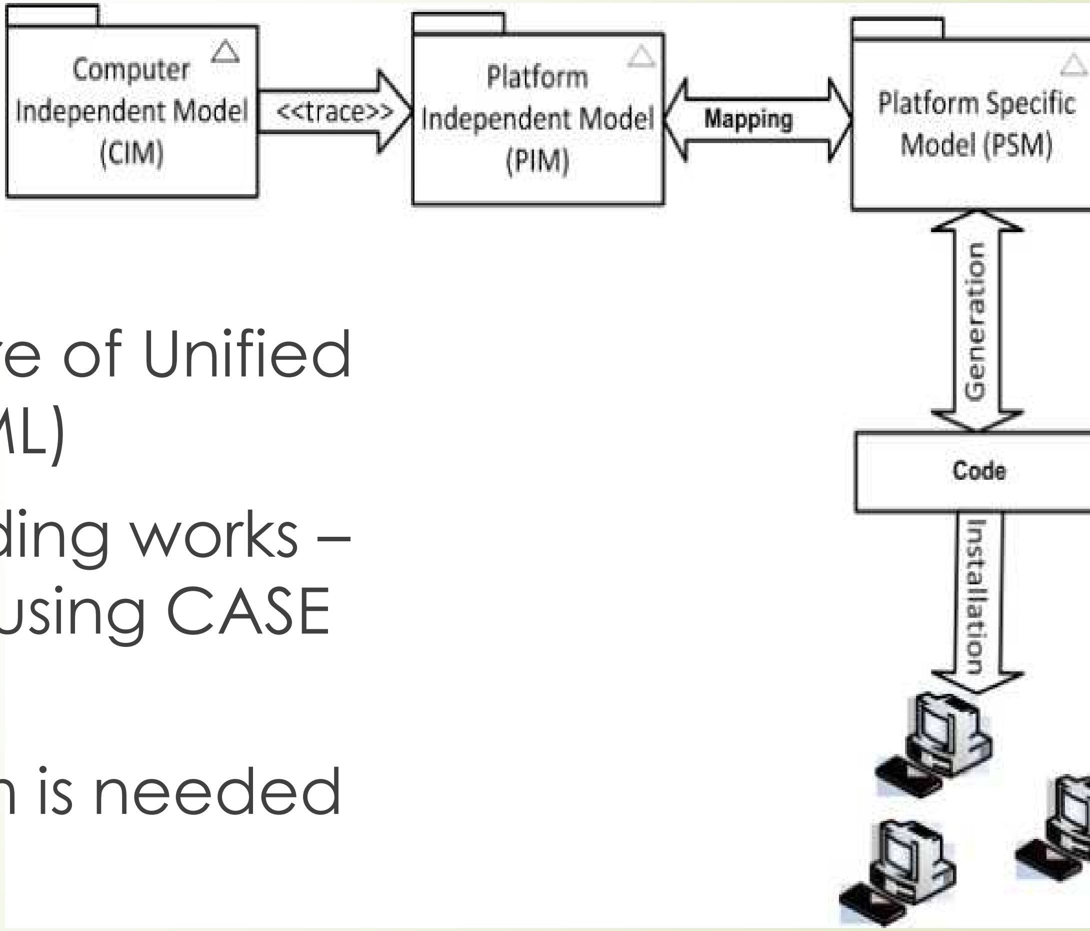
- Modelling in software development process
- MDA Approach in software development process
- Agile Approach in software development process
- Comparison of MDA and Agile approach
- Differences from the comparison
- Advantages and disadvantages brought by the defferences
- When to use traditional approach (MDA)?
- When to use agile approach?
- Possibility of combination of these approaches

# Modeling in Development Process

- Most important phase in traditional methodologies
- Way of information (about being developed software) visualization and exchange
- Way of information documenting (for future need)


Software Development Team Member
Software Development Team Member
Software Development Team Member
Project Manager – Solution Provider
Models of IS
Software Development Team Member
Software Development Team Member

# Model Driven Architecture in Development Process


Computer Independent Model (CIM)
<<trace>>
Platform Independent Model (PIM)
Mapping
Platform Specific Model (PSM)
Generation
Code
Installation

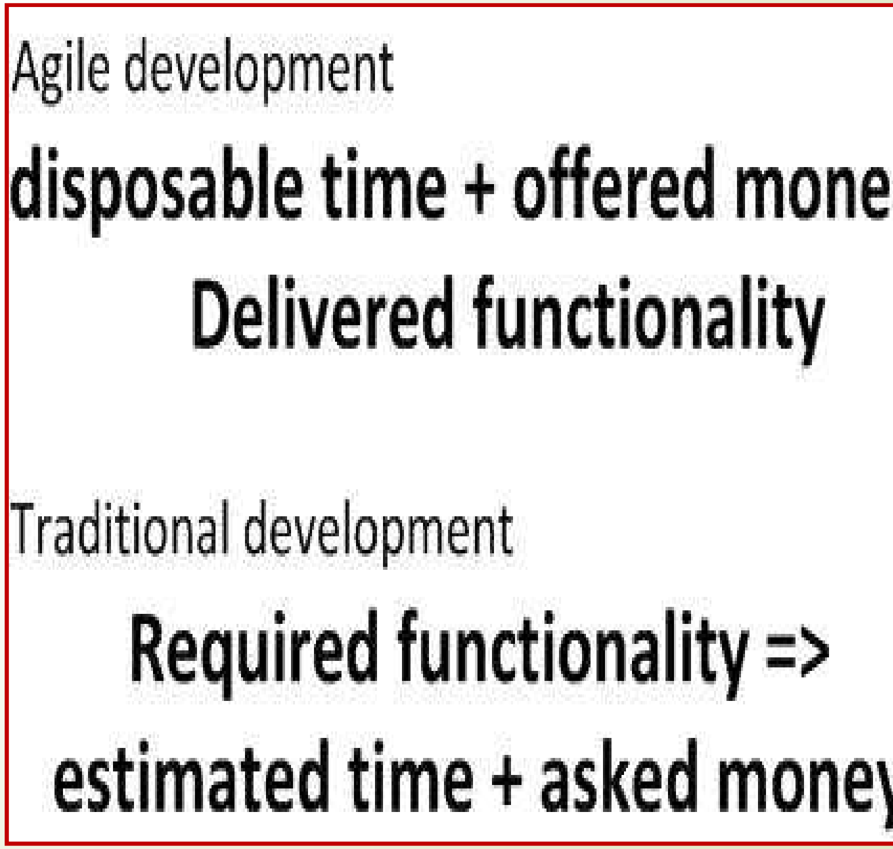
- Supposed to be the future of Unified Modelling Language (UML)
- Idea: minimization of coding works – models are transformed using CASE system
- Compatible CASE System is needed

# MDA Models

- CIM – Computer Independent Model
  - System requirements
  - Project Dictionary
  - Main System Functionality
- PIM – Platform Independent Model
  - Models with all important details
  - Algorithms of the functionality are modeled
- PSM – Platform Specific Model
  - Components of the platform are included in the models
  - Code level details are visualized

# Agile Approach - Basis

- New approach for managing software development project
- Minimizing modeling works
- Rapidly increasing programming works instead
- Minimizing documentation



Agile development

**disposable time + offered mone**

**Delivered functionality**

Traditional development

**Required functionality =>**

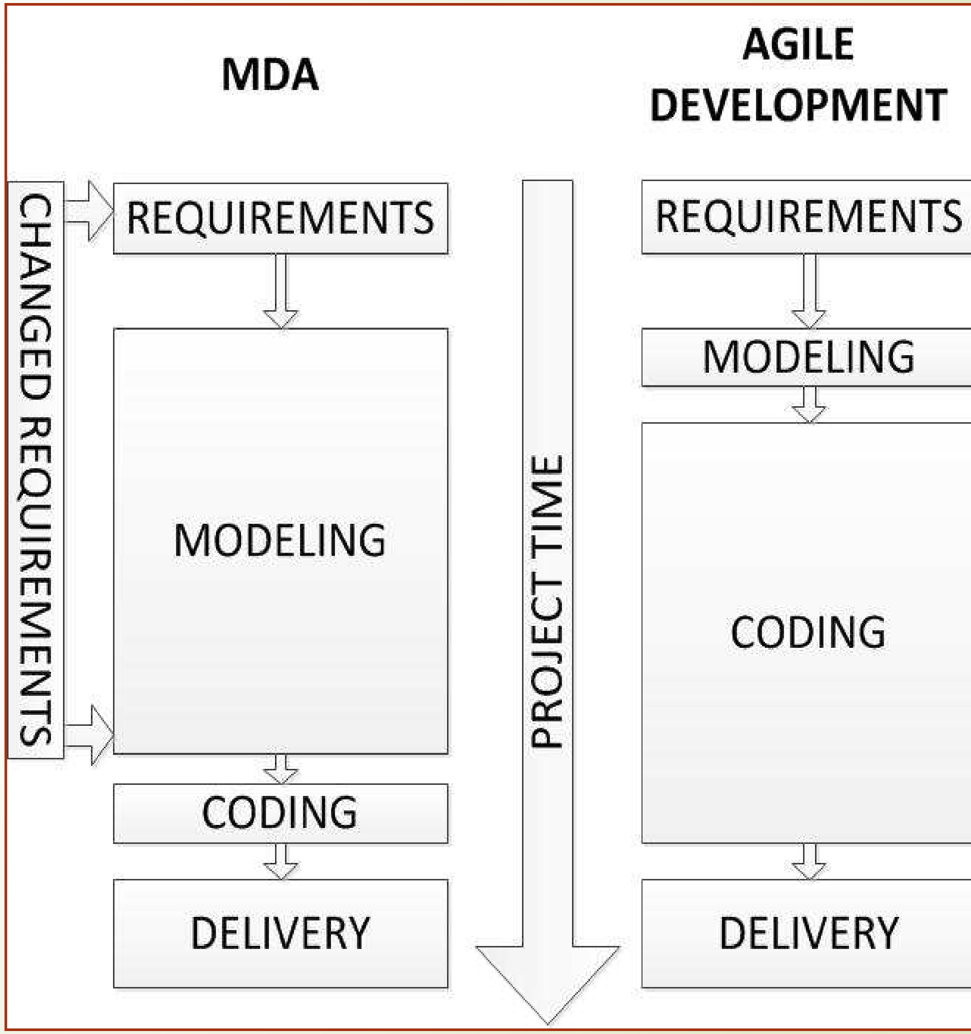
**estimated time + asked money**

# Agile Approach - Characteristics

- Methodologies supporting this approach are widely used, for example SCRUM
- Intensive communication with the customer is needed – because the models do not realize the documentation
- The first functioning part of the software is delivered as soon as possible in spite of the risk of recoding the works (when it is not done according to the customer's wishes)
- Important information with the need of exchange between the customer and the development team and between the members of the development team as well is not visualized, is only given by word

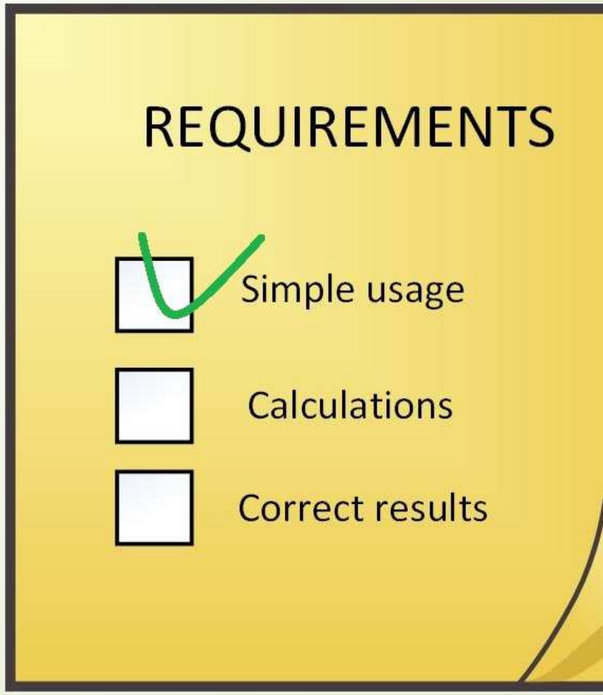
# MDA and Agile Development – Comparison - 1

1. Amount of modeling and coding works (MDA – modelling represents most of the works, in agile coding is realized mainly)



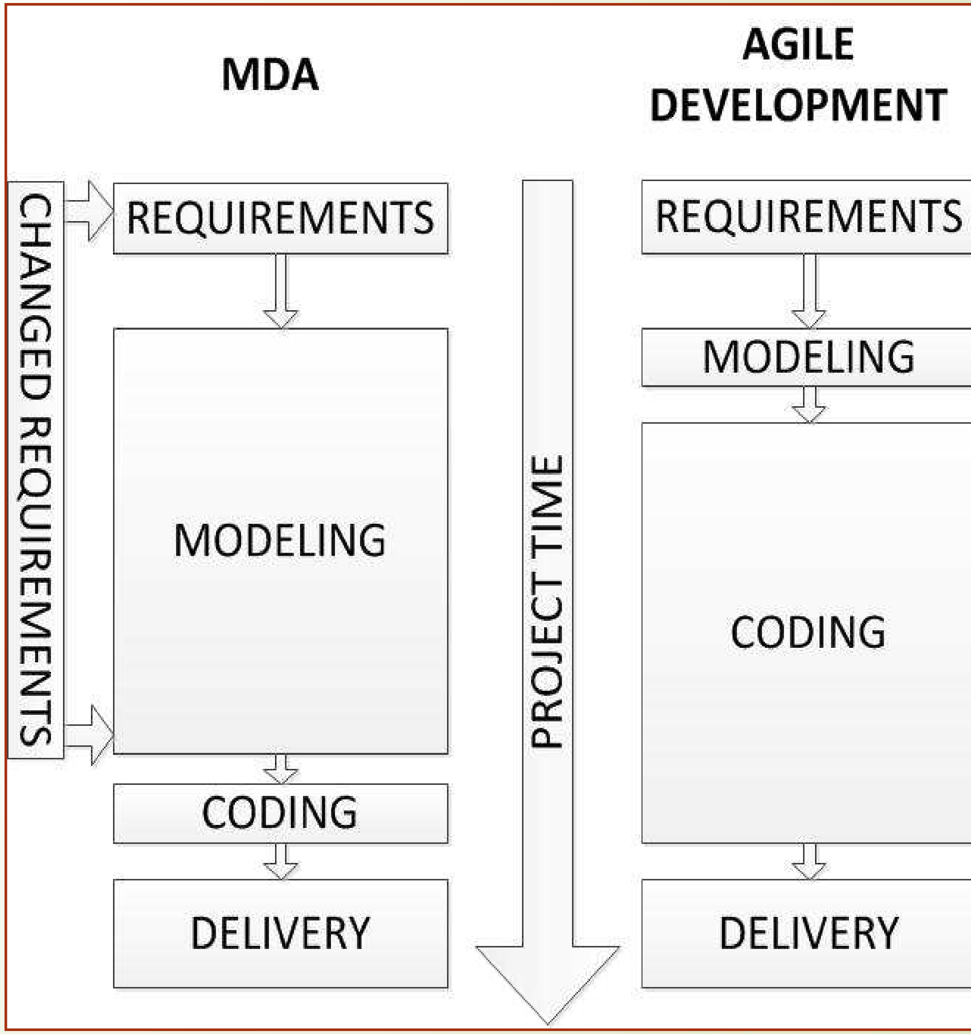
# MDA and Agile Development – Comparison - 2

2. Change Management – no matter which approach and methodology is used for the project, the requirements will always be changed by the customer.
   As the software is being built the customer has to more specify his needs and can realize that some details should be done other way as he has specified before.



# 2. Change Management

- MDA – changed are solved in the models
- Agile – changes are solved directly in the previously coded parts

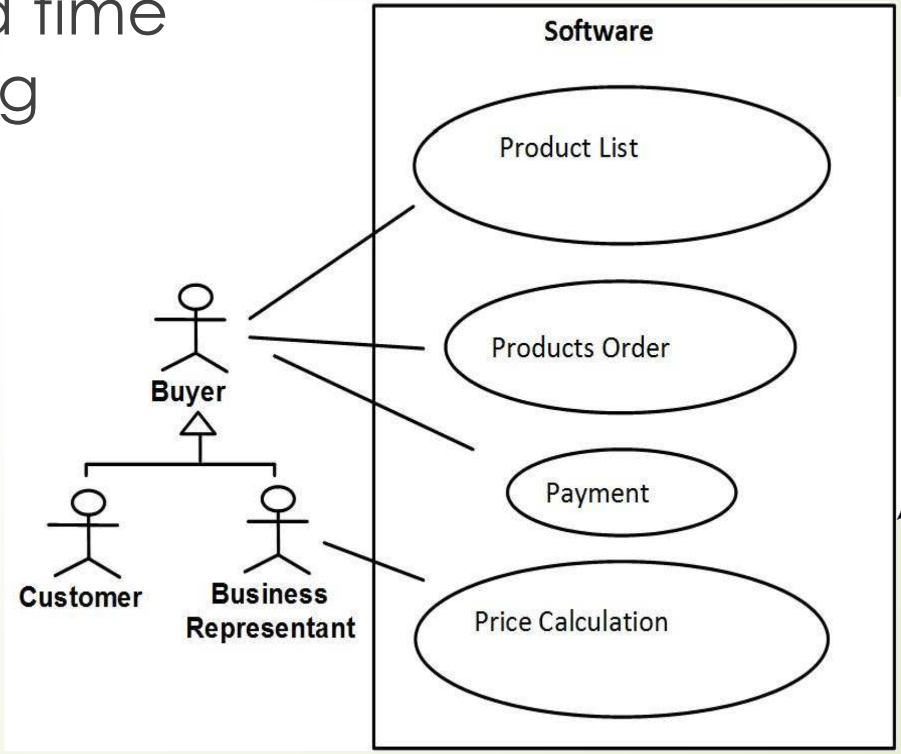


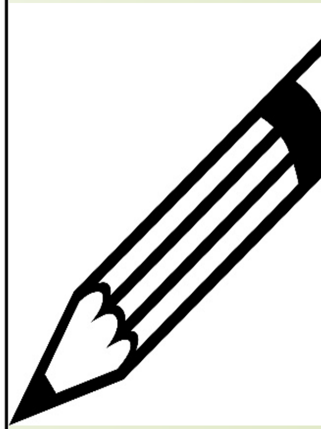
# Advantages of modelling the functionalit

- clearly defined information being shared gives more possibilities for cooperating of more people on solving one problem
- Information are captured in the models creates strong tool for requirements stability. Once the customer has agreed with the solution – it is documented and can not be ignored later
- Keeping the important information for future need – very often it happens that something needs to be added to previously developed system and the information about previously realized solution are important than. The team members easily can not be present and if the information is captured it is easier for the new members to discover the previous solution

# Disadvantages of modelling the functionality

- Time spending – creating models needs time and time means spending money





# Disadvantages of modelling the functionality

- ‘Nothing to show to the customer’ – when models are created and discussed the customer can not work with the new software, it does not exist yet.
  It seems sometimes as the solution provider did nothing in spite of the fact that the team members solve the problems and visualize the accepted solutions in this time

# Advantages of requirements modelling

- Advantages of requirements changes solving in the models:
  - Time spending in case of bigger changes. If the changes are more complex, then the solution finding process is more effective when reworking the models than reworking the code
  - Changes with the influence to other project parts. Some changes can require changing other parts. Modelled changes help to better recognize the dependencies.
  - Documenting the changes. In some cases the history of changes is needed.

# Disadvantages of requirements modelling

- Possibility to see the change for the customer. If the change is realized direct to the code, the customer can see the new version much sooner.
- Time spending in case of elementary or not complex changes. Modelling these changes only spends time, reworking the code directly could be much faster

# When traditional approach?

- Big development team with various nationality, age and geographically distributed (different working time – direct online communication is not easy)
- Geographically remote customer or customer is hard to communicate directly because of any other reason
- Not clear idea of the future system on the customer side, high risk of requirements changing
- Complex software is to be built, with more problems, which must be solved and exact software algorithms defined
- Complex software with expectations of future additional functionality delivering

# When agile approach?

- Well coordinated development team, easy to online communicate
- ‘modern type of customer’ – easy to online communicate, with disposable time for communication about the future system
- Simple software, clearly defined by the requirements
- Simple, minimum changes during the development process

# Combination of traditional and agile approach -1

- When it is possible to divide the tasks as more complex ar less complex:
  - Must be no or very minimum works dependences between these two groups
  - Must be no or very minimum works dependences between the tasks in ‘simple’ group
- The more complex tasks are modelled and documented. The solutions are found in models, requirements and their changes are documented

# Combination of traditional and agile approach - 2

- Less complex tasks are coded (in case of need recoded) directly and the customer can see this part of the project immediately
- Important to have each task clearly grouped so as each team member can always know which task he works out and which approach to apply (model or code directly)

# Conclusion

1. MDA – Model Driven Architecture represents traditional approach
2. Agile approach – modern approach with goal to lower the needed time for development
3. Both of approaches bring some advantages and disadvantages depending on their usage for the softwar development project
4. Good choice which approach to use for with project is c strategic managerial question and can influence the project success
5. Combination of these two approaches is possible in specific types of projects so as to take the advantages c both approaches and avoid the disadvantages as well

# Thank you for your attention

- Questions