

# Climbing the MDSD mountain (1)

- **Objectives** of MDSD
  - *Flexible* implementation: platform changes
    - *Derivation* of different PSM is possible
  - Simpler and more effective *maintenance*
    - Changes can be done directly to existing designs
  - *Effective* development: Common language; requirements *traceability*; earlier *testing* and *simulation*
    - *Separation* of concerns: allow stakeholders to be focused on a specific domain
    - Reduces the *loss of information* from logical to technical implementation
    - Model (*conceptual*) integration is easier than application integration
    - Improves requirements traceability: changes and validation
    - Facilitates early testing and simulation



# Climbing the MDSD mountain (2)

- **Objectives** of MDSD
  - Increased *productivity*: automation; increased reuse; reduction of rework
    - *Automate* steps of the development process
  - *Quality* improvement
    - Reduces the amount of *rework* due to errors
  - Updated *documentation* of the system
  - Ensure customers, designers and architects *understanding*



# Climbing the MDSD mountain (3)

- **Difficulties** of adopting MDSD
  - Shift in development *culture*; staff not ready for modelling; new rôles are needed
    - Requires people to be *trained* in modelling: analysts vs. programmers
  - Difficult to distinguish *real* MDSD/MDA providers
  - Lack of *confidence* on MDSD/MDA promises being real



# Climbing the MDSD mountain (4)

- **Difficulties** of adopting MDSD
  - Usually seen as a *heavyweight* methodology
    - High importance of *maintaining* the modelling approach (instead of tweaking the code)
  - *Transformations* promises not a reality yet
    - Strong dependence on *quality* of models and transformations
  - Incomplete and not interoperable nor integrated *tool chain*
  - Relatively *high cost of adoption* (training, infrastructure, tools)
    - Requires the development of basic “*infrastructure*”
  - Definition of an extension mechanism to allow customization and specialization without breaking the code generation

