



Software Center

▶ **Continuous model-based *software* development and lightweight model consistency checking**

Jan Carlson, Antonio Cicchetti,
Federico Ciccozzi, Robbert Jongeling

ModProd 2019-02-06





Background



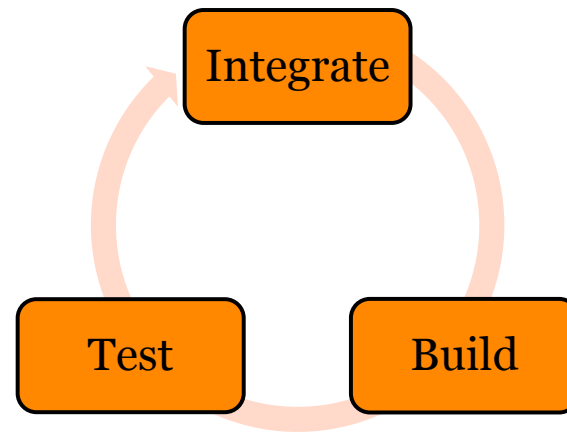
- Growing software (and system) complexity
 - Also in domains where software traditionally played a smaller role
 - Highlights the need for development at higher abstraction levels
 - Use of models for documenting, communicating, analysing and implementing software
- Need for shorter development cycles and faster feedback
 - Addressed by Agile development processes
 - Strive to avoid heavy upfront design
 - Focus on working software over comprehensive documentation
- Agile Model-based Software Development
 - Reconcile continuous development and modelling practices





Continuous Integration

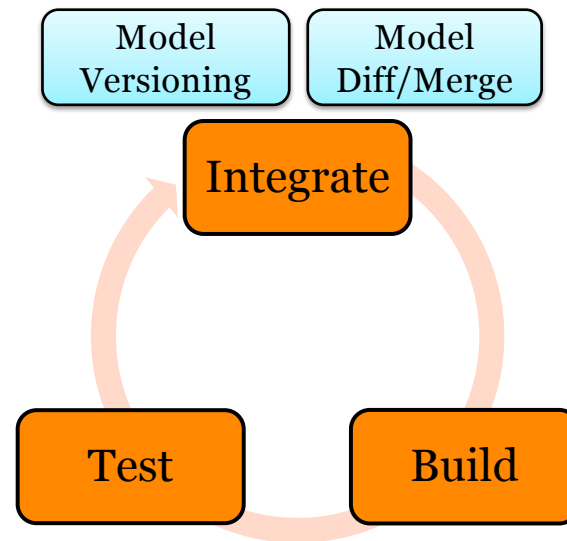
- One of the agile software development practices
 - Frequent (daily) integrations into a shared repository
 - Automated builds
 - Automated unit (and integration) tests
- Well-established practice at code level





Model-based Continuous Integration

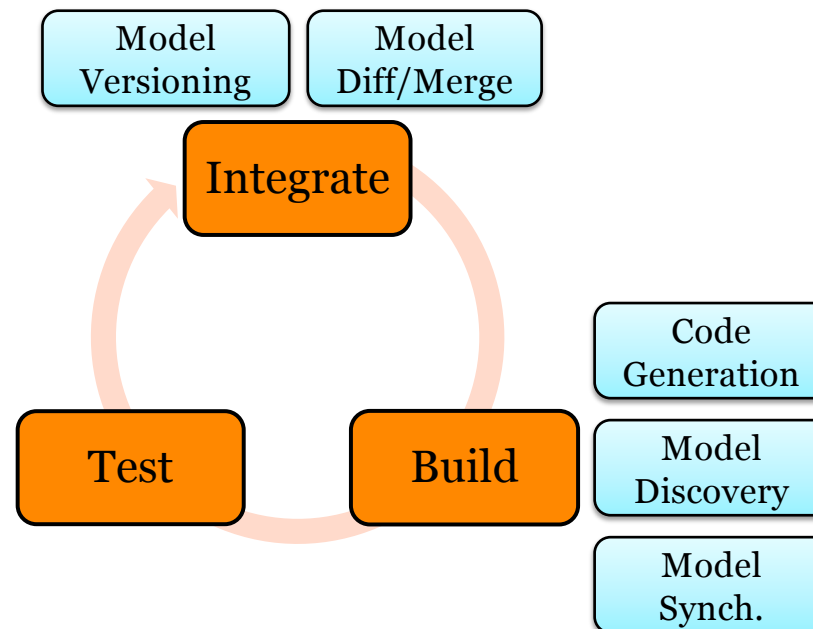
- Integration
 - Model-aware version control
 - Differencing and merging at model level





Model-based Continuous Integration

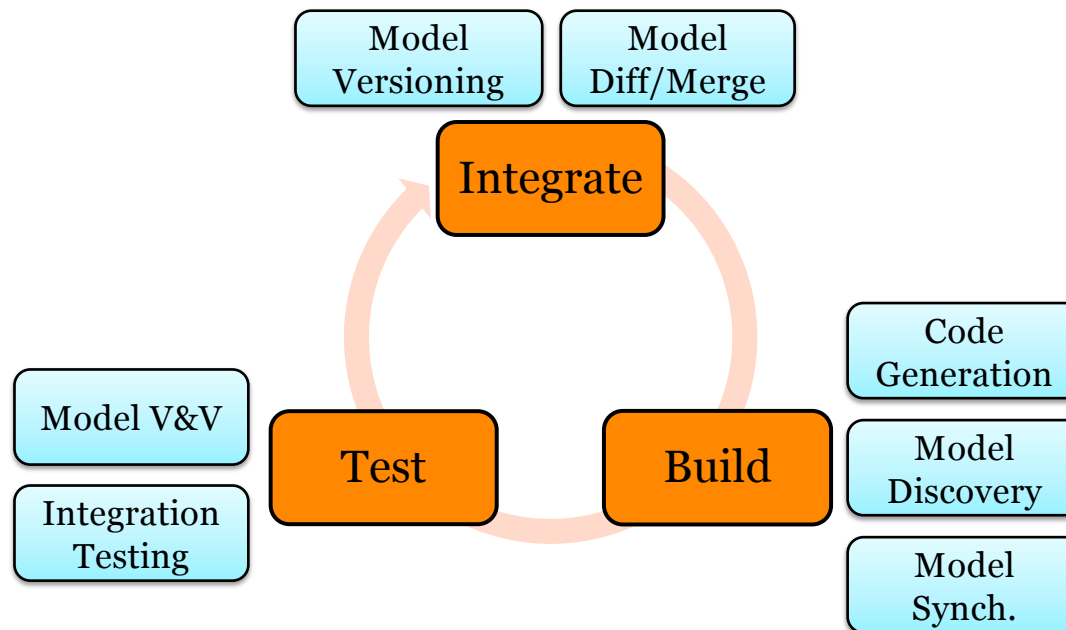
- Building
 - Incremental code generation from models
 - Model generation from code changes
 - Inter-model synchronization (especially with multiple modelling tools)





Model-based Continuous Integration

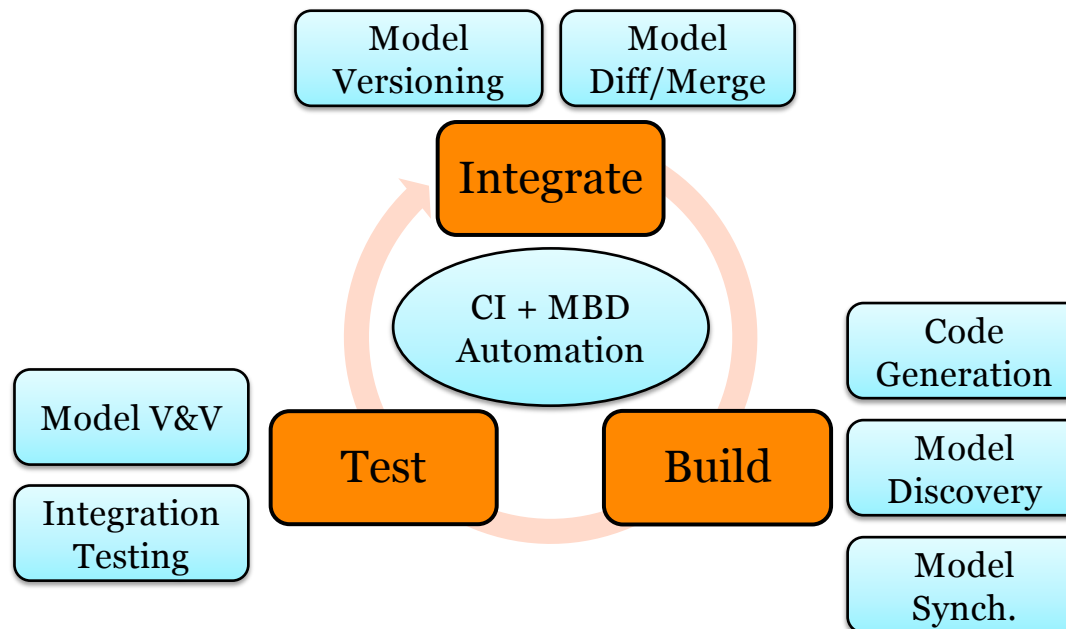
- Testing
 - Model testing & model-based regression testing of generated code
 - Integration testing (especially with multiple modelling tools)





Model-based Continuous Integration

- Automation
 - Minimize the need for manual interaction in all steps





Impediments to Combine CI and MBD

- Tool review
 - 8 tool (focusing on UML/SysML and Simulink)
 - Reviewed with respect to the identified relevant aspects
- Results (presented at the COMMitMDE workshop @MODELS'18)
 - Diff and merge at model level in most tools
 - Versioning by integrated version control (e.g. svn or git)
 - Management of generated code varies
 - Support for automation and customization varies
 - Main challenge when multiple tools are used together



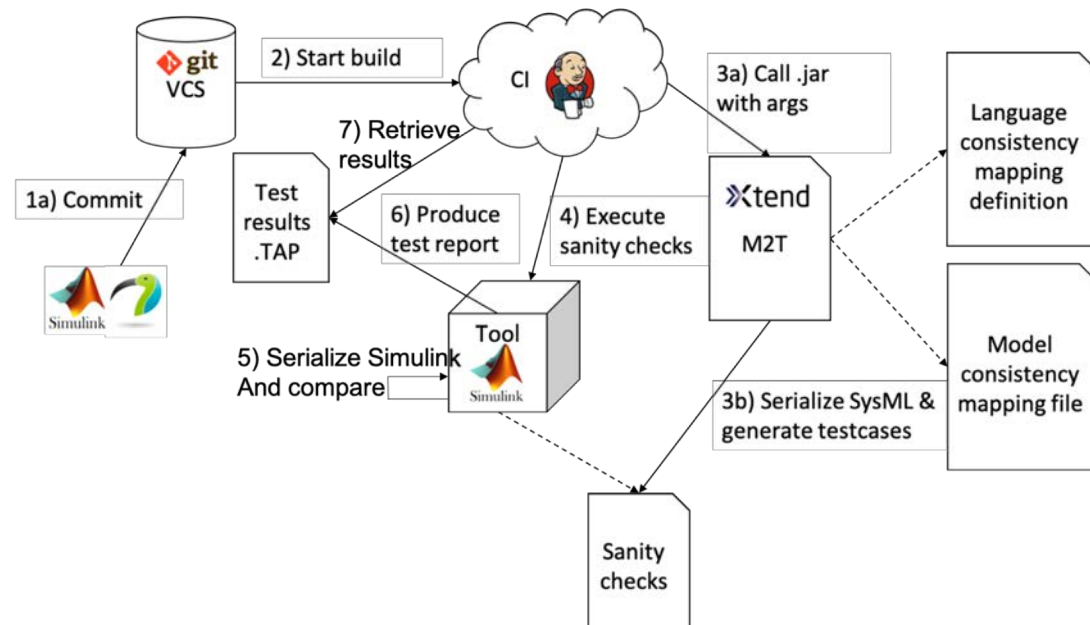
Impediments to Combine CI and MBD

- Interview with practitioners (ongoing)
 - 12 practitioners, from 3 companies of different modelling maturity
 - Interviews addressing
 - Current MBD + CI practices
 - Views on further CI and/or more modelling
 - Perceived impediments
- Preliminary result: Few impediments reported
 - Modelling only at system level to document design decisions, no models used at software level
 - Modelling only at software level, with complete MDD tool chain and code generation (no manual code)
- Remaining challenging scenario (?)
 - Modelling at multiple levels, in multiple languages, with multiple tools



Lightweight model synchronization

- Inter-model (and inter-tool) consistency checking
 - Automated – integrated with the Jenkins CI pipeline
 - Lightweight – warn the user about potential inconsistencies
 - Incremental – build the consistency mapping as the models evolve
 - General approach, initially targeting Simulink+SysML





Thank you!



- Please contact us if you are interested in these topics or if you want to know more about our research!
 - Jan Carlson <jan.carlson@mdh.se>
 - Antonio Cicchetti <antonio.cicchetti@mdh.se>
 - Federico Ciccozzi <federico.ciccozzi@mdh.se>
 - Robbert Jongeling <robbert.jongeling@mdh.se>

