
# Model-based Design for Safety Critical Controller Design with ROS and Gazebo

Michael Naderhirn, Mischa Köpf, Josef Mendler

# Background (as I understand it)

- Conformity means that a development was done according to safety critical standard
  - ISO 26262 – Automotive
  - DO-178C, DO-330-333 – Aeronautic
  - IEC62061 – Machinery
  - If “something bad” happens companies must show that the development was done conforming to the relevant safety critical standard
  - An independent auditing company helps a company through auditing that the development processes conform e. g. DO-178C
- Certification means that an authority certifies that a development is allowed to be used
  - UAV is allowed to fly into the civil aerospace (e. g. flies according the rules of the air)
  - Autonomous car is allowed to drive on the road (e. g. drives according the rules)
  - Conformity can be part of the certification
- Qualification means that a tool is qualified to be used for safety critical development
  - Documentation that a code generator generates correct code – e. g. tests show that the generated code produces the right and wrong results correctly
  - An independent auditing company audits the software used by company, the methods to test the software and the documentation

# Product Development Cycle

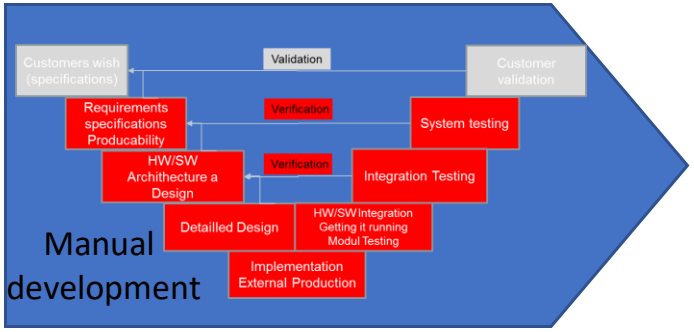

Technology-phase

Feasibility
↓
Technology selection
↓
Proof-of-Concept

Predevelopment-phase

Concept
↓
System design
↓
Prototype

Serial-phase



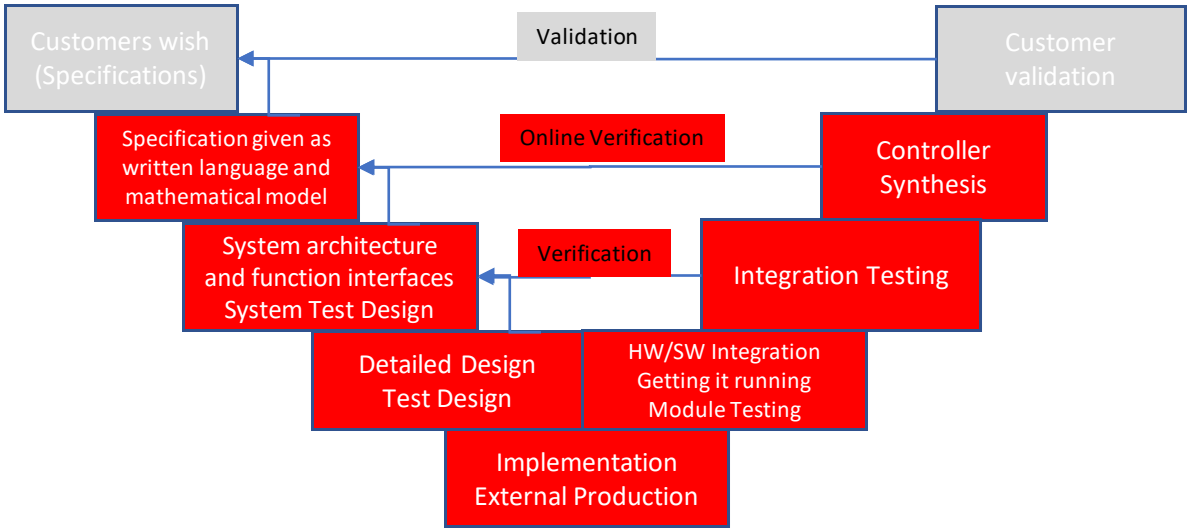

MATLAB ROS

OpenModelica

Know How after predevelopment-phase

Often not used

# Basic Design Steps




- Defining the system spec
- Hazard and risk analysis
- Determine Safety Integrity Level (SIL)
- Define controller structure and necessary redundancies

- Develop tests for the system, subsystem and modules
- Develop and test modules
- Integrate modules with subsystems and test
- Integrate subsystem with system and test
- Validate system with customer

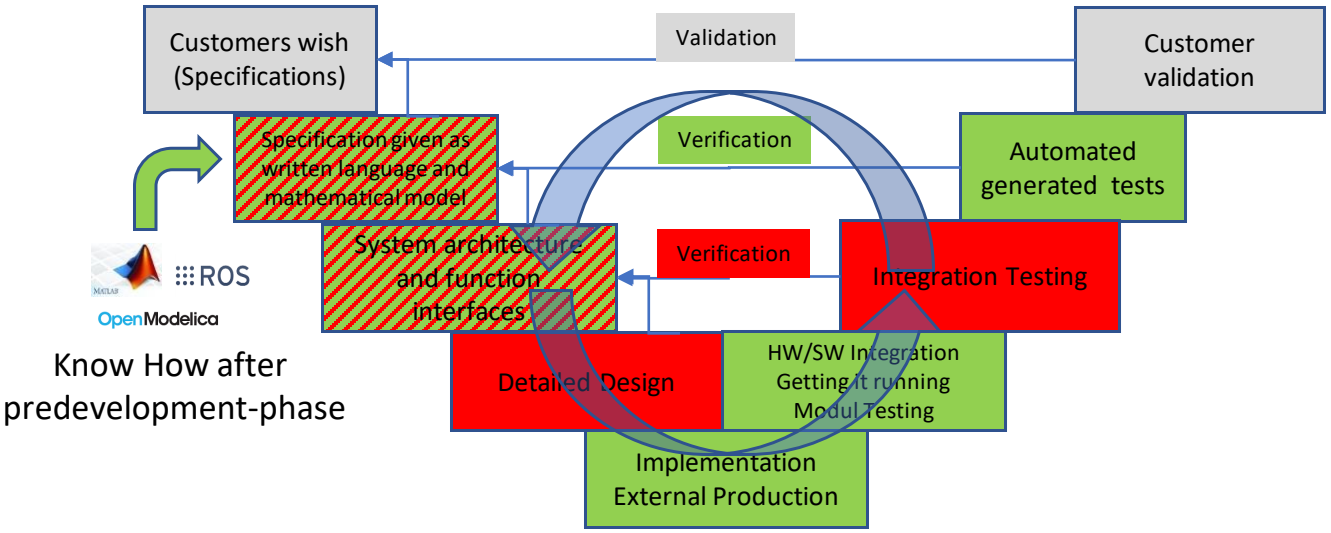
# How to get better?

- New mathematical methods to automize development process
  - On- and Offline system verification
  - Verified deep neural networks

- Listen to customer (some important findings)
  - Development in 90% preferably done on Windows computers
  - ROS is used in app. 80-90% of prototyping of robotic solutions
  - SMEs have problems to afford commerical development tools – some stop robotic projects after prototyping
  - Robotics engineer needs to be a software archithect
  - Available knowhow and packages should be reusable

  -> High degree of automation required

# Incremental (Agile) Development

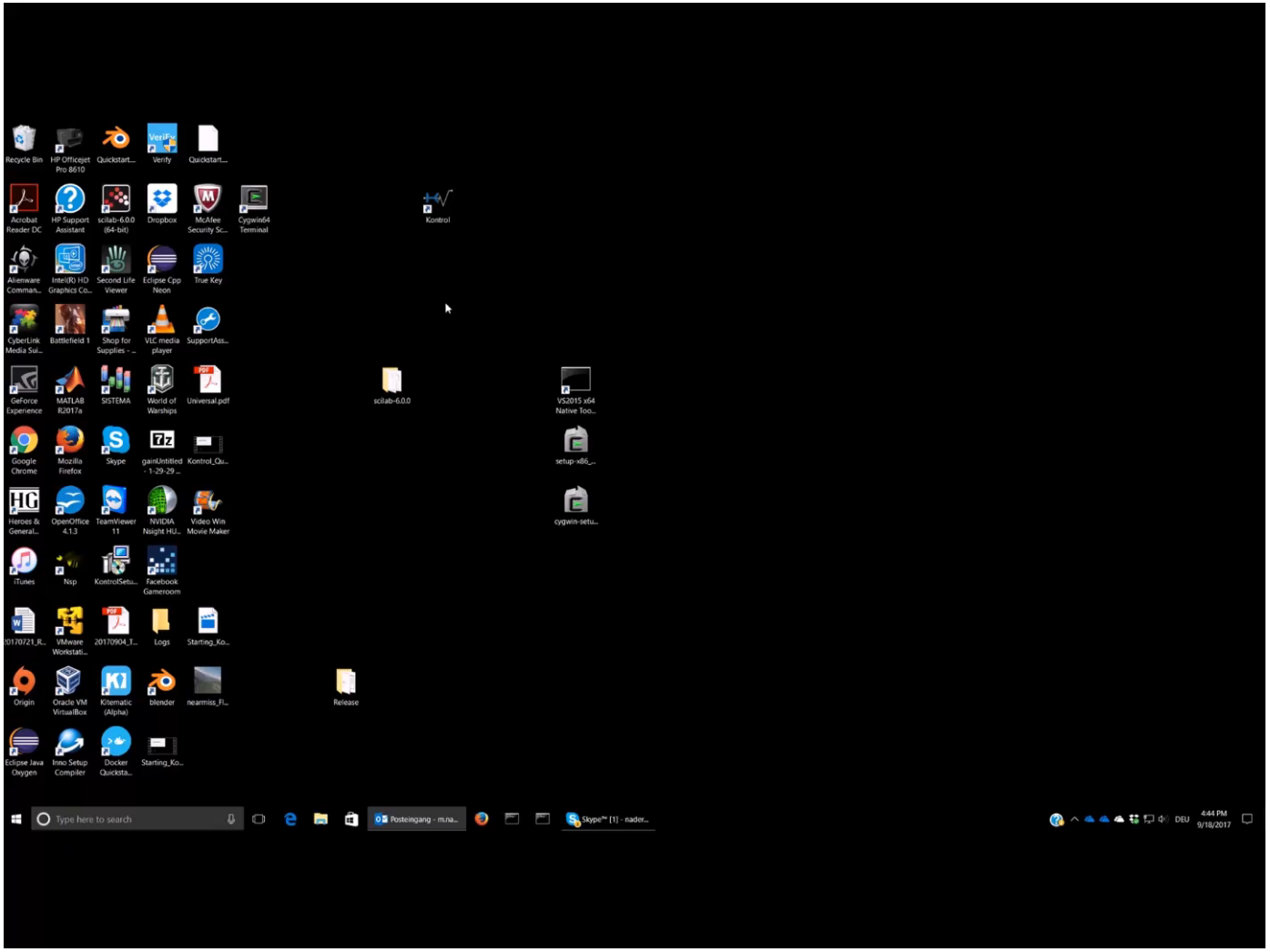



Agile development of mechatronic systems allows faster time2market

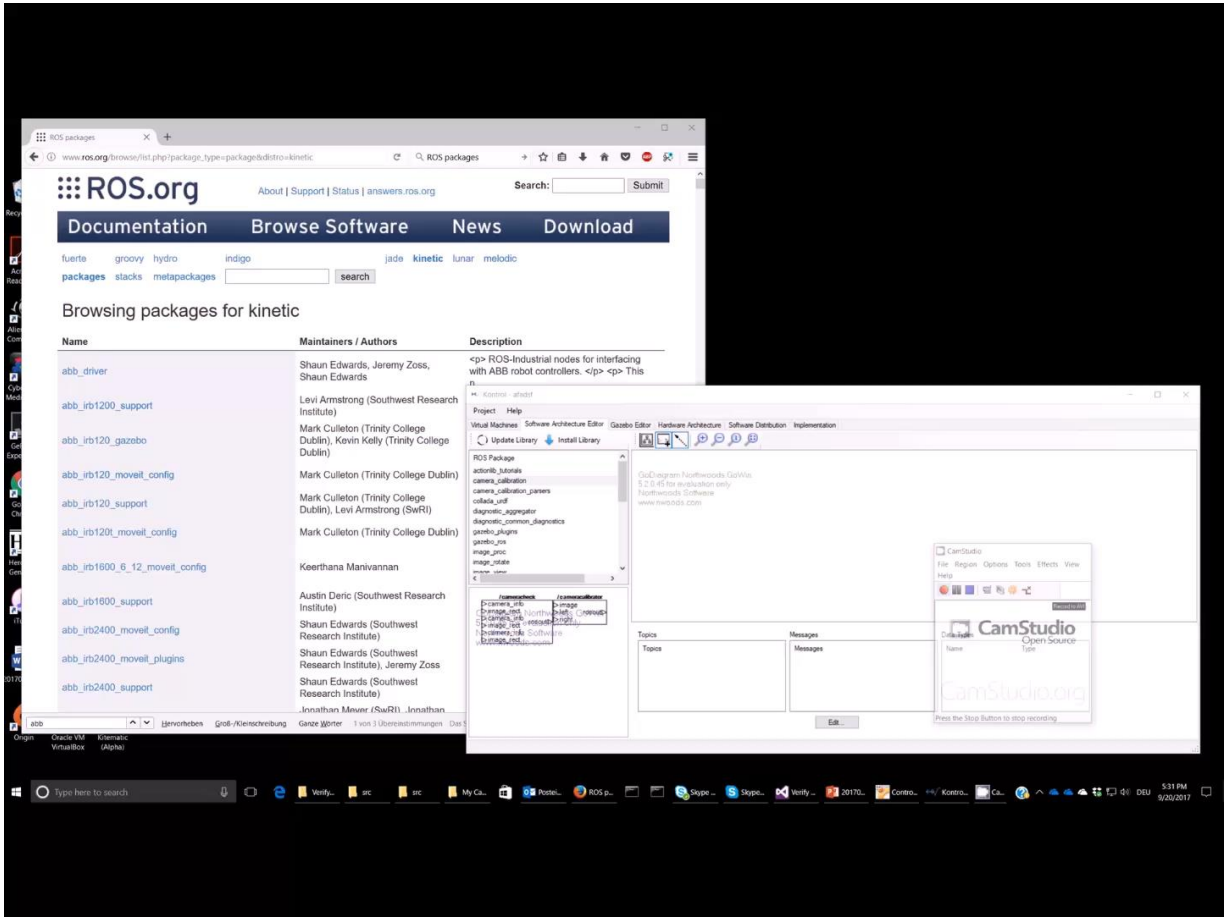
# Our Solution - Kontrol

- Integrates
  - ROS / ROS2
  - Gazebo
- Imports ROS packages and prototypes automatically
- Model based design of ROS structure
- Software distribution to hardware and automatic configuration
- Manual coding or model based design
- Allows to add and edit Gazebo world
- Scilab connects to ROS network automatically from Windows
- Generates Code for ROS (and ROS2)
- Independent from simulation environment
- Future – Implements an incremental development process for safety critical controller design for mechatronic systems

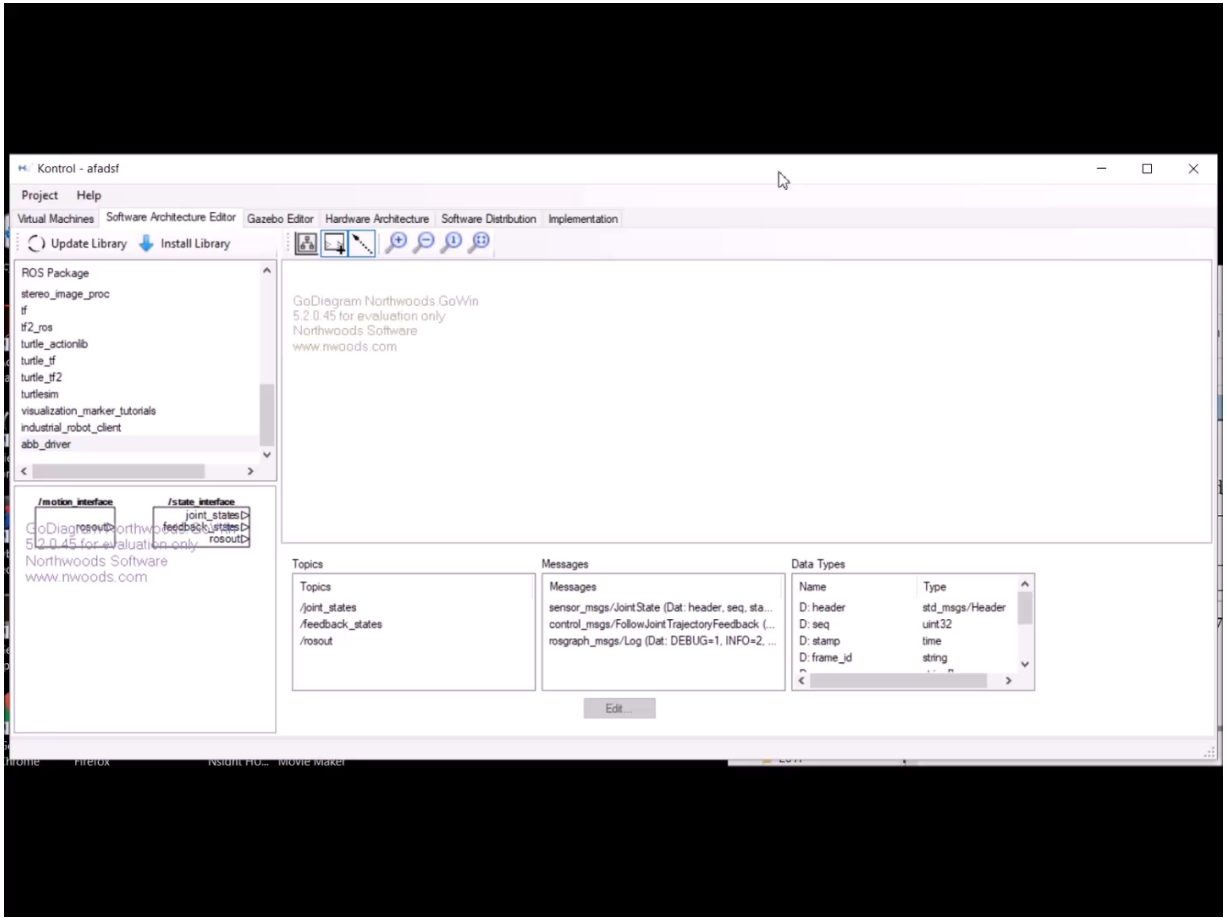
# Our Solution - Kontrol



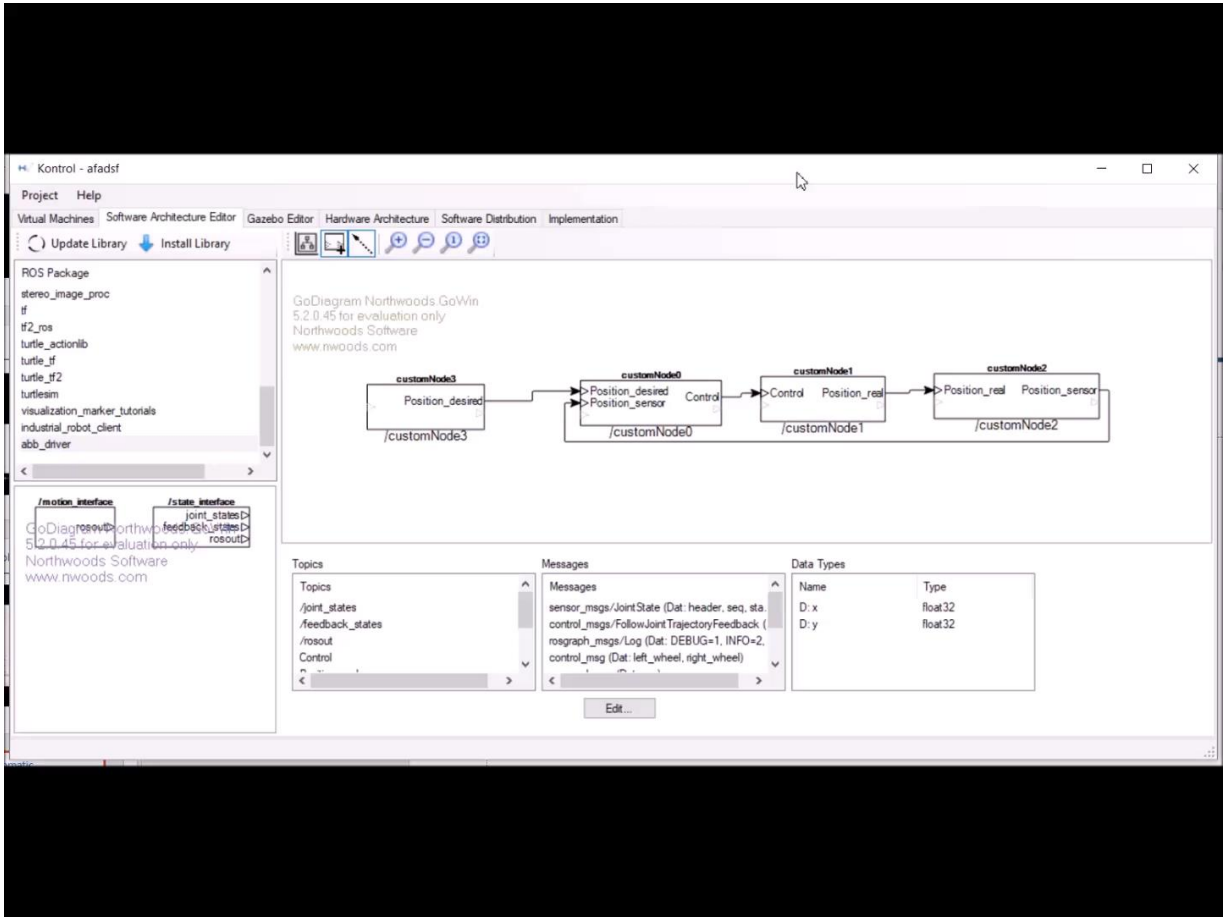
# Import of ROS Packages



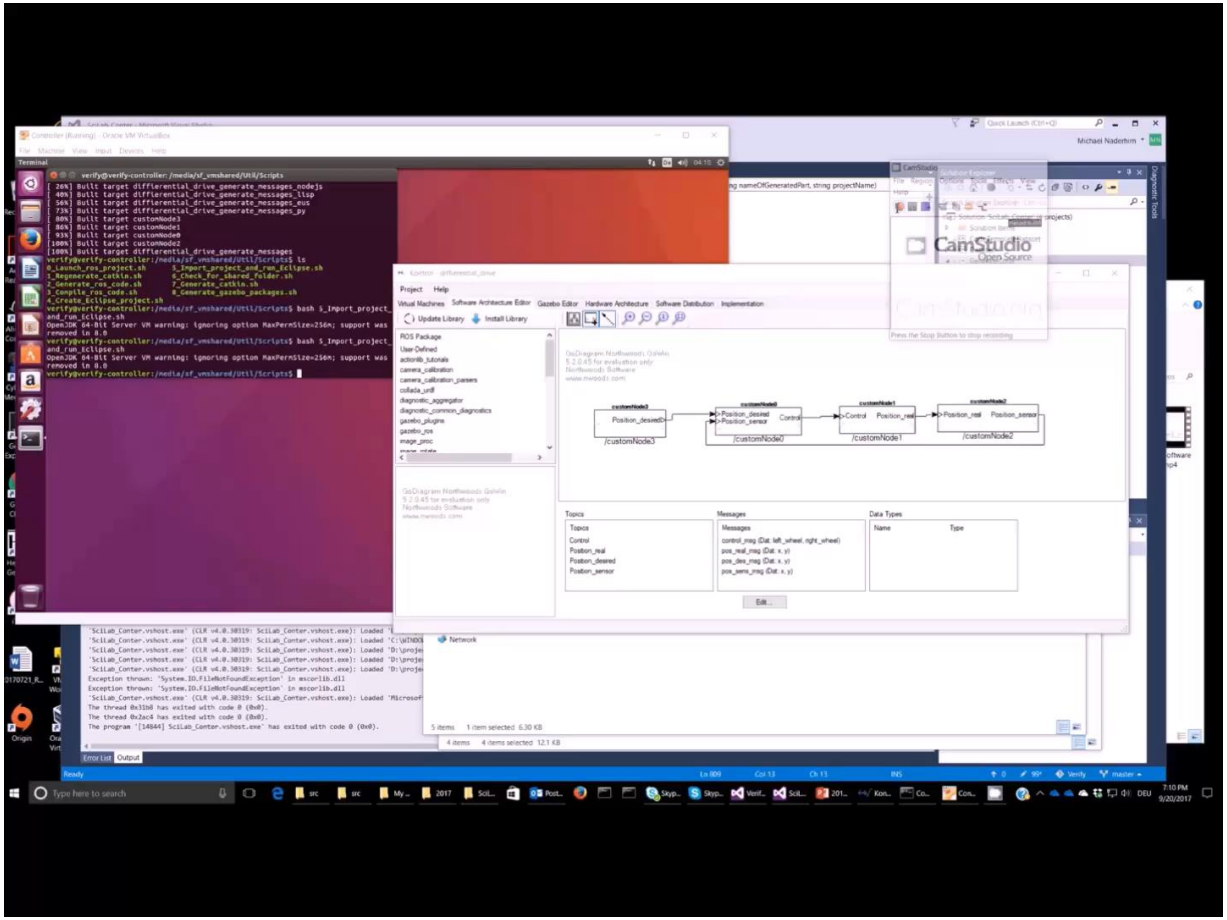
# Model-based Design of Controller Structure



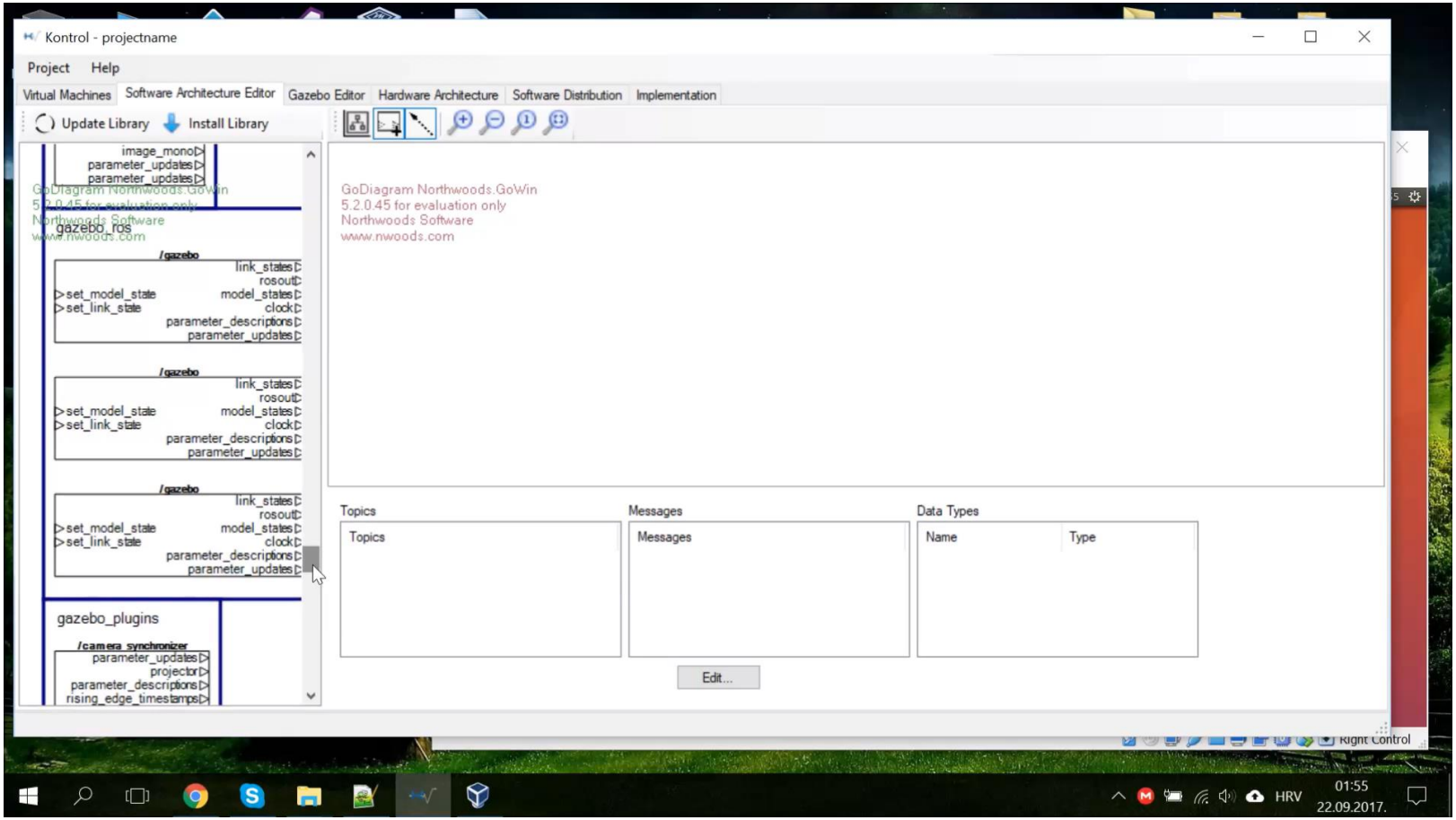
# Software Distribution and Automatic Configuration



# Manual Coding and Model-based design



# Add and Edit Gazebo World



# What we want? / What we are looking for?

- We are looking for Beta Testers
  - Register at kontrol.tech – Beta testing
  - Will start in Q4/2017

# Contact

Kontrol GmbH
Dr. Michael Naderhirn
Marienhöhe 25
4391 Waldhausen
Austria
Email: m.naderhirn@kontrol.tech
+43-676-9760483

Kontrol GmbH
Dr. Josef Mendler
Marienhöhe 25
4391 Waldhausen
Austria
Email: j.mendler@kontrol.tech
+49-175-5738846