

# Transition to Autoware.Core/Universe

Ryohsuke Mitsudome

March 9, 2021

# History of Autoware



AUTOWARE.AI

- + ROS1
- + various packages combined to realize AD stack
- No architecture design
- limited documents and test codes



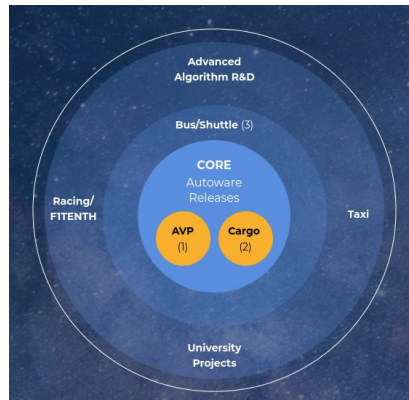
- + ROS2
- + rewriting the code from scratch
- + Defining target use case (AVP, cargo delivery, etc)
- + Design architecture
- + writing design documents and test codes
- High entry barriers for new engineers
- Making large change is difficult
- no platforms for experimental packages

# Core

- Complete end-to-end autonomous driving framework
- Supports all current AWF ODDs
- Provides the definitions and functionality for which other packages can extend
- Strict code and quality control
- Heavily managed by the AWF
- Stable base implementation

# Universe

- Additional packages built on top of Core
- Extends Autoware functionality beyond the AWF ODDs
- Completely dependent on Core functionality and message definitions
- Relaxed code and quality control
- Community managed
- Enables quick experimentation and prototype testing



# Our goals

- To build safe and high-functioning Autonomous Vehicles efficiently.
  - It doesn't mean to accomplish "Automotive Grade" right now.
- (After that) To heat up Autonomous Driving community/business with Autoware.

There are some challenges for this.

- We cannot know the entire requirements/specifications from the beginning.
- Generally it's difficult to accomplish both development speed and quality at the same time.

To solve this:

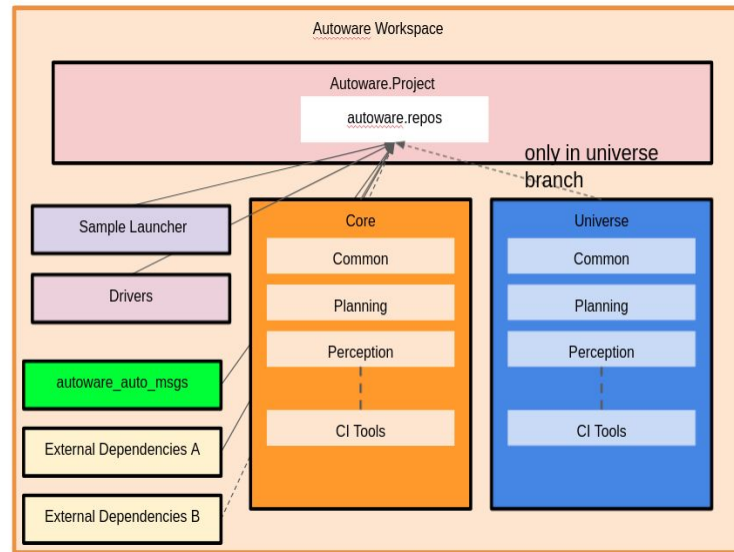
- We should have prototype-development phases.
  - Use case based development under light rules for rapid prototyping.
  - High-quality development under strict rules for production level code.
- Actually, we're doing similar process as ODD based development such as AVP/Cargo/Racing/Bus/...
  - However, the current Autoware.Auto development seems to be mixing prototyping and production.
  - That slows down development speed or incomplete code is merged to production codebase.

# Repository Structure

- Starting from a simple structure for Bus ODD and will revisit it later.
  - One .repos
  - One branch(Only universe for now)

Key repositories ([see here](#) for the details)

- **Autoware:** contains repos files
- **Core:** stable packages (currently empty)
- **Universe:** experimental packages(currently Tier IV proposal)
- **Common:** common libraries (e.g. cmake)
- **Launcher:** launchers for demos
- **Documentation:** basic documentation for autoware



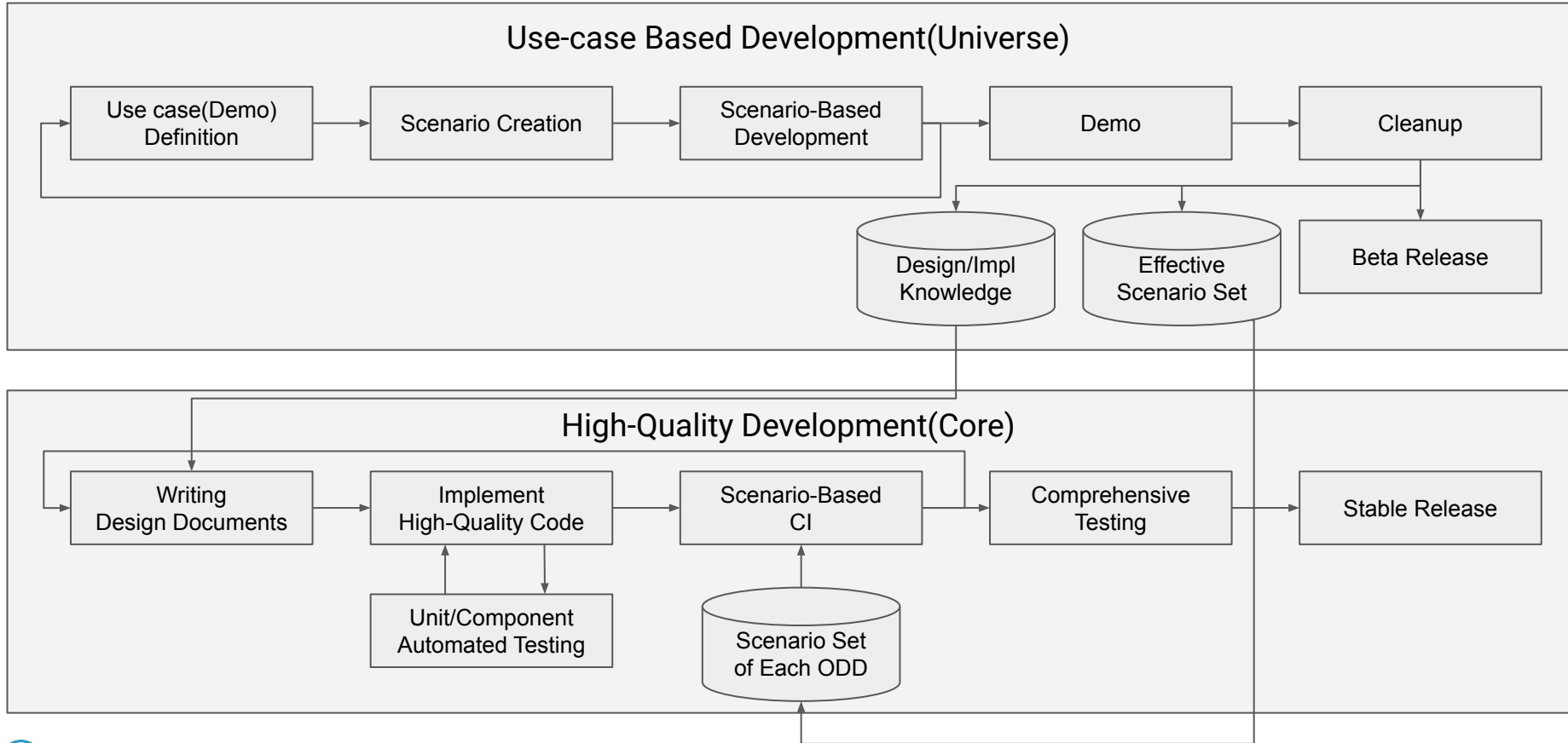
# Quick Start

- [Installation instruction](#)
- [Sample Launch](#)
  - Set initial pose and goal pose to test planning
  
- More detailed documents available soon in autoware-documentation
  - Current [PRs](#)

# Introduction to Github tools

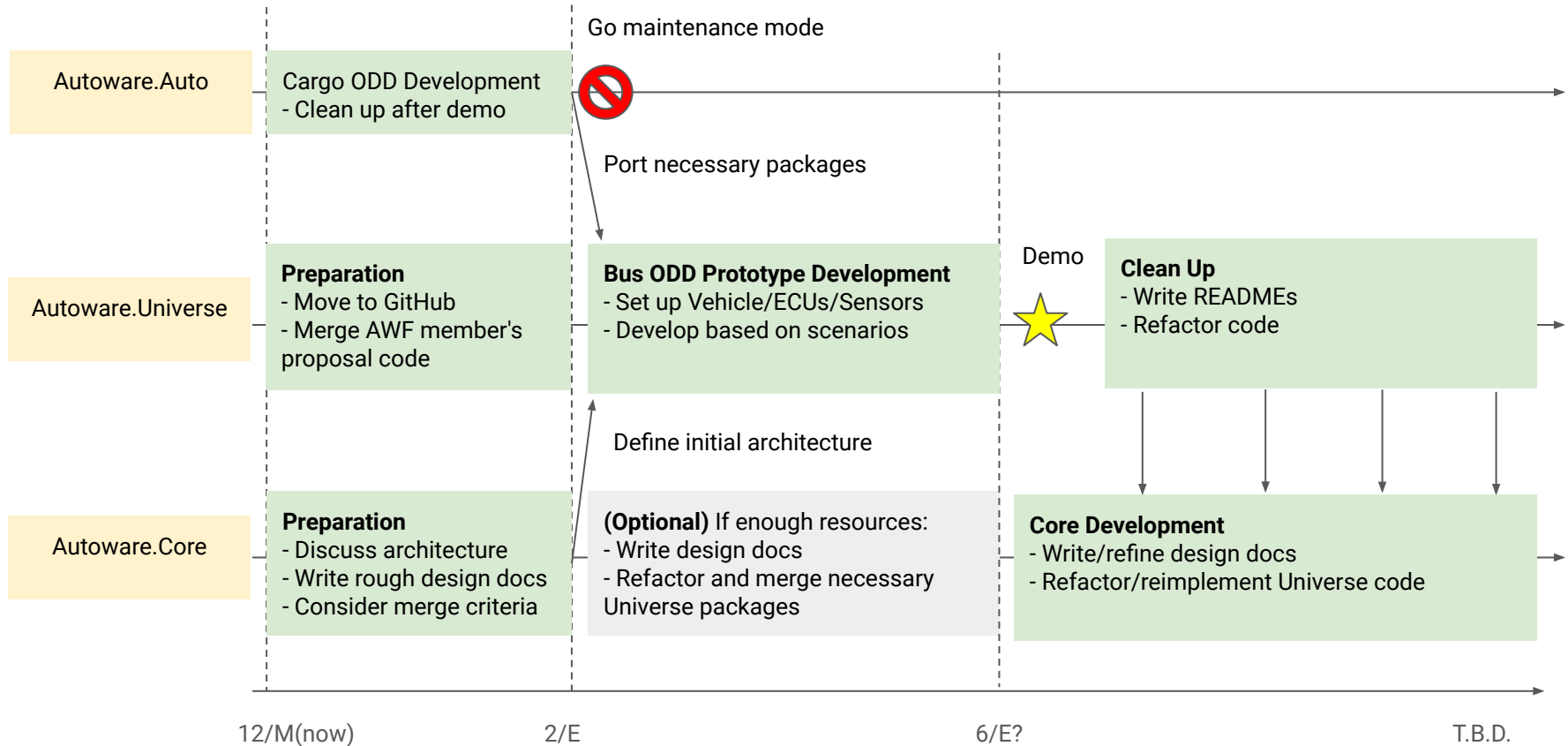
- [Projects](#) for managing WG tasks
  - equivalent of [Milestones in Gitlab](#)
- [Wikis](#) for working groups
  - Ported from Gitlab wiki
- [Discussion threads](#) for discussion
  - We will be using issues for tasks and bug reports

# High-level development process of Core/Universe





# Timeline



# Task status

Category	Task	Status
Cargo ODD Development	Clean up after demo.	Finishing up
Universe Preparation	Move to GitHub and build the base development platform.	Tier IV is working on at <a href="https://autowarefoundation.org/autoware.universe/">autowarefoundation/autoware.universe/</a> .
	Merge AWF member's proposal code.	Call for participation. Possible port some packages from .Auto
Core Preparation	Discuss software architecture.	Ongoing in <a href="#">Discussion: Bus ODD Software Architecture - Autoware.Auto</a> . We should call for participation to this as well.
	Consider merge criteria including coding guidelines, test strategy, etc.	Tier IV is preparing a draft of proposal. Will propose by the January TSC.
Bus ODD Prototype Development	Many (e.g. hardware specs, setup milestones, etc.)	