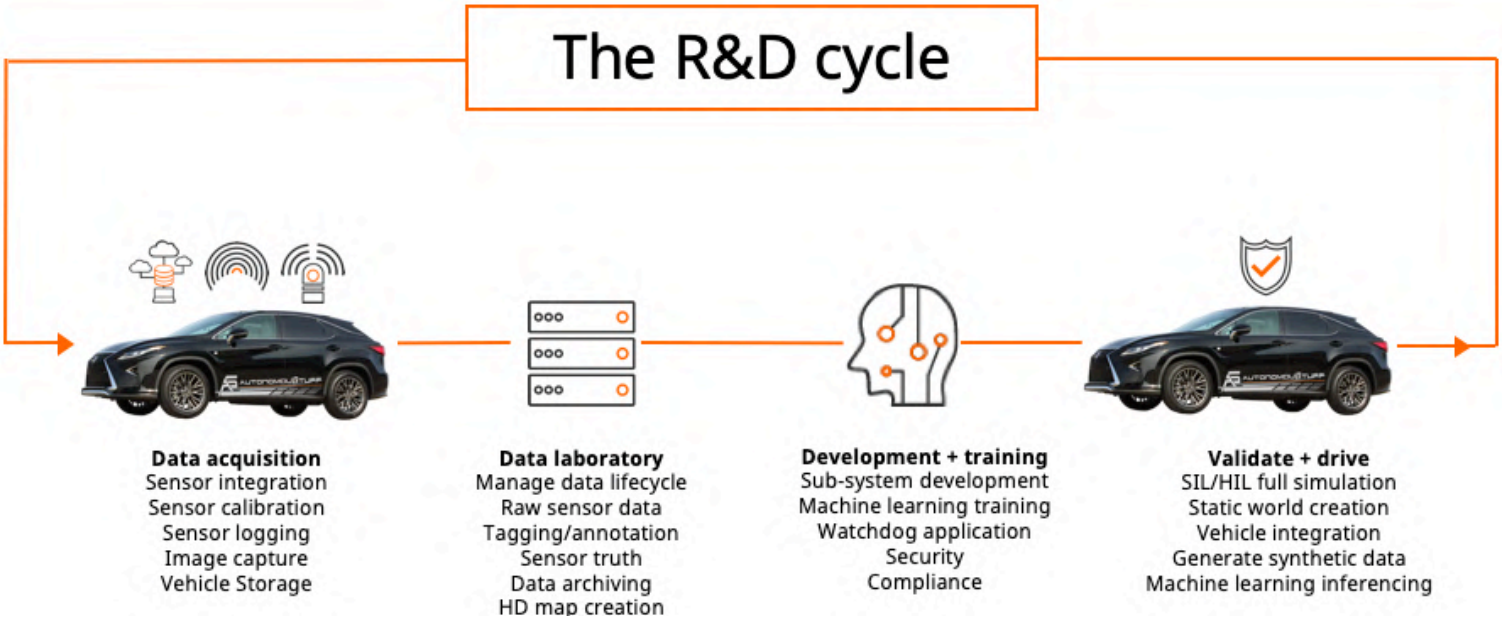
Data Intelligence

White paper

The role of data in autonomous vehicle development



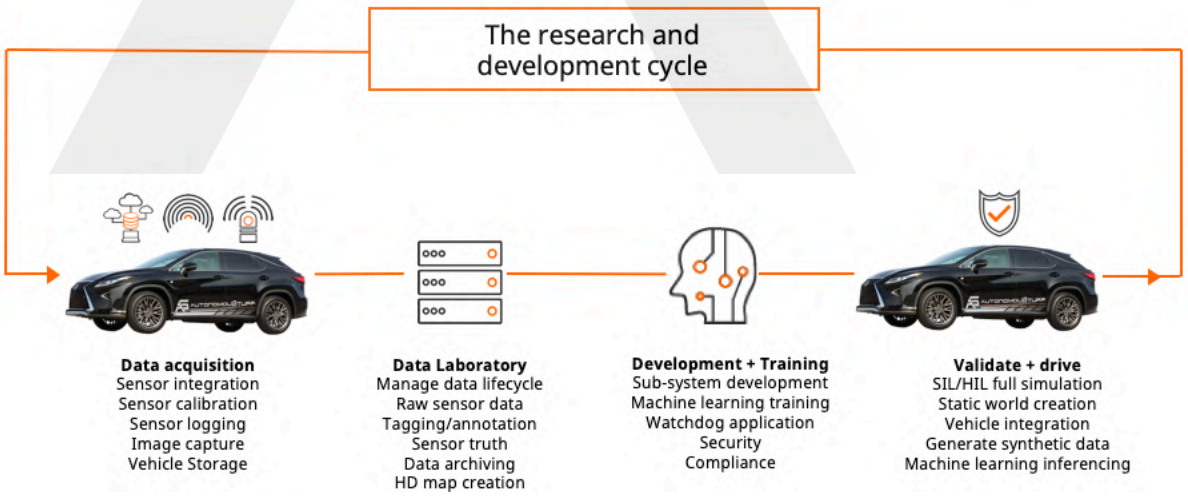
Lee Baldwin, Director of Data Intelligence
AutonomouStuff

Background

In the context of developing autonomous vehicles, Data Intelligence (DI) is the focus on vast amounts of data generated by highly automated vehicles. DI concentrates on efficient ways of capturing, transferring, processing and archiving the data from autonomous research and development vehicles. DI also focuses on the development of the autonomous vehicle software stack that controls the vehicle and the tools needed to enable the development of the stack. These DI tools consist of storage, annotation, AV map creation, computing, machine learning model training, machine learning inferencing, and simulation.

Data Intelligence pillars

The following diagram depicts the R&D cycle that is typically followed when developing an autonomous vehicle. The cycle below is repeated thousands of times, and as R&D programs mature, it becomes more important to economically and efficiently perform each of the pillars below. This white paper explores each of the pillars and highlights some of the more important aspects that are often overlooked but require a well-thought-out strategy.

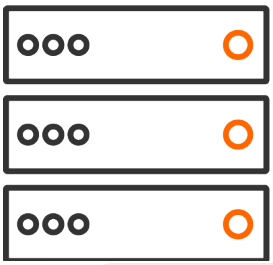


Data acquisition

Autonomous vehicles generate typically between 8 and 20 Terabytes (TB) a day — and in some cases can generate 20TB in a single hour. When an autonomous vehicle is developed, there is a strong focus on sensors, integration and development of the software stack, but not on efficient vehicle data capture, data transfer to engineering teams, and data retention. Engineering teams are focused on building a solution, and data acquisition tasks are typically left up to information technology (IT) organizations to sort out later, when the R&D program matures. Involving IT infrastructure personnel up front so that they understand the requirements of the engineering team is critical to selecting an appropriate solution. Typical unstructured data requirements of the past do not apply to R&D autonomous vehicles due to the volume of data created, so understanding the volume and retention of the data is critical to choosing storage technology. The following list of factors, though not comprehensive, plays into selecting the most appropriate storage solution:

1. How much data will the vehicle sensors create each hour and how long will the vehicle be out on a mission before it can offload the data?
2. What is the maximum ingest rate of all of my sensors? This determines the type of in-vehicle storage — spinning disk, SSD or NVMe.
3. How long do I have to copy the data from the vehicle to where my engineers need to access the data? This drives whether you need removeable storage, can utilize a direct data connection to the vehicle, or need to upgrade your internet service for transfer to the cloud or other locations.

4. How many vehicles will offload data after a mission, and will it be simultaneous? This will drive network infrastructure and on-premise storage requirements.
5. Will my engineers be using a cloud provider for development?
6. If you plan to utilize the cloud, then how much data will be uploaded? One TB will take a minimum of two and a half hours over a full 1 Gb connection. Does the cloud provider have an offline upload technology where secure drives are sent to the cloud provider?
7. Do I need to keep raw sensor data, and if so, how long do I need to keep it? What is the intended archive tier — slow disk, tape or cloud?
8. Do I need a very fast file system for my data scientists for machine learning training?
9. Who will I involve from my IT organization, and how close are they to the decision maker?



Data laboratory

The data laboratory is where the data from the autonomous vehicle is stored, managed and manipulated.

Part 1: Storage lifecycle management

Regardless of whether the data is stored on-premise or in the cloud, the data needs to be stored on the appropriate storage tier. The tiers are defined as hot (tier 0) to cold (tier 3). Tier 0 is ultra-high-speed storage (NVMe or SSD); Tier 1 is high-speed storage (SSD or disk); Tier 2 is slower speed storage (disk); and Tier 3 is archive/object storage (disk or tape). The hot storage tiers are typically the most expensive, while the cold is for long-term retrieval and very economical. The determination of how much data needs to be stored on the appropriate tier must be made up front in order to choose the most economical storage solutions. Typical cloud providers do not charge for uploading data to cloud storage, but storage and retrieval charges can rack up, so it is important to understand the intended data use cases. For example, if a data scientist plans to utilize the cloud for machine learning training, then the cloud is likely the right platform. However, if the data scientists work on local deep learning workstations, then moving data back and forth to the cloud will be inefficient. The best practice is to define high-level storage requirements, including the forecast of data growth and intended use of the data. Then the solution that has the lowest total cost of ownership can be identified. More than one storage solution may be needed to address unique requirements.

Part 2: Annotation

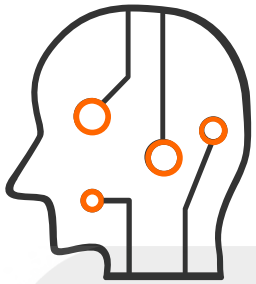
Once sensor data is stored in the cloud or in an on-premise storage solution, the data will need to be filtered and reduced to a manageable subset for annotation. Filtering the data is important, due to the volumes of data generated from the vehicle sensors. For example, two high-definition video cameras capturing 37 frames per second amounts to 1.3 TB of data per hour and would generate more than a quarter million images. Assuming that images have been down selected and point cloud data in 10-second intervals for interesting events (pedestrian interaction, intersections, ramps), then this data needs to be annotated so that machine learning algorithms can be trained. Typically, data scientists will note scenarios that they are looking for, so sensor data can be collected and fed into the machine learning training. It is important to find an annotator that is economical, but more importantly is accurate for the type of annotation required. The typical types of annotation are bounding boxes, semantic segmentation, and point cloud annotation. Data privacy is becoming more important, so understanding whether data anonymization is required also factors in to choosing an annotator. Data accuracy is very important, as annotated images are utilized to train the autonomy perception sub-system. Bounding boxes around objects and the placement of the box is very important for machine learning training. Inaccurate bounding boxes can lead to inaccurate image classification, resulting in vehicle perception systems making unsafe choices. Other considerations for annotators: the ease at which the images or point cloud data is uploaded to the cloud provider; the use of AI for pre-classification (they tend to be faster and more consistent annotators); whether the labor force is crowdsourced or third party (annotation consistency and privacy); availability of references for direct communication, and dedicated project management resources for large jobs.

Part 3: Autonomous vehicle high-definition mapping for driving and simulation

One of the necessary items for autonomous driving is a current, highly accurate map, with precision down to better-than-10-cm accuracy. The map must be current so roads, road signs, construction detours, signals, buildings and other objects or landmarks are where they are in real life. Currently, the high definition mapping industry and format is like the wild west. There are many providers and many map formats for vehicles. When developing an autonomous vehicle, it is important to be agnostic regarding map formats, since this market is currently very turbulent. For the time being, the best choice is a mapping company that meets AV stack requirements and will map any designated area — or choose to conform to the mapping provider's specification, such as HERE or Tom Tom and locations that they have mapped.

A second map format, such as an OpenDrive format for simulation software, likely will be needed. This map is used to create a digital twin of the real-world testing area in the simulation software. In order to build out the simulated world, another software package typically is required. That software requires a map file so the static scene elements such as buildings, signs and parked cars are placed on the map.

Another considering for a map provider is "localization" and how the autonomy planning system will localize the vehicle on a high-definition map. Autonomy planning sub-systems need to know where the vehicle is on the map verses reality. Popular methods are GPS coordinates or overlaying point cloud data. Some map providers provide a localization sub-system, some just provide a map, some will provide a map and a LiDAR point cloud that corresponds to the map. The point cloud generated by the autonomous vehicle is overlaid on the stored map point cloud to localize the position on the map. These are just a few things to think about when choosing a mapping provider.



Development (application) + training (machine learning/AI)

There are a few full autonomy software stacks such as Tier 4s Autoware, Baidu's Apollo and the Nvidia Drive platform, but most startups, OEMS, Tier 1s and silicon providers are either developing sub-systems or are developing a full proprietary stack. The major sub-systems being developed are planning and perception. Perception is used to determine what the autonomous vehicle is "seeing" so that it can make a decision on what to do next, and planning is related to determining where the vehicle should go next. The perception sub-system interacts with onboard sensors such as high-definition/4K cameras, thermal cameras, radar, LiDAR and ultra-sonic sensors to determine what objects are around the vehicle, including pedestrians, moving vehicles, parked vehicles, traffic signs and signals, curbs and other road conditions (pot holes). The planning sub-system will typically utilize camera images, GPS and point cloud data with a high-definition map to accurately "localize" and determine where to go next based on conditions and the mission objective (for example, going to the grocery store).

Engineering development teams are faced with developing the software required to have the sub-systems interface with vehicle systems. The onboard sensors that these teams utilize also feed data to onboard machine learning (AI) for perception and planning systems. Data scientists are the experts in machine learning and have the knowledge on how to configure and train machine learning algorithms. Machine learning algorithms get "smarter" as they are fed information in "training" mode, which is typically done where the engineers are located. The machine learning computers are typically high-end GPU workstations or servers. When perception machine learning algorithms are trained, annotated images are fed to it in "training mode." These perception algorithms require hundreds of thousands

of images before they are “smart” enough to drive safely on real roads. One use case is feeding annotated images of pedestrians, where the annotated image has all of the pixels that represent pedestrians noted. After the machine learning perception algorithm is trained with the annotated pedestrian images, then the algorithm is placed in simulation or in a vehicle for testing and the machine learning algorithm is set to “inference” mode. Inference mode is where it is digesting sensor input where the perception system classifies images. This classification occurs by comparing what the sensor “sees” to what it has “learned” and assigns a probability to each object in the image. This process must occur in real time for driving. There are startups that are specializing in predicting what pedestrians will do, which illustrates the complexity of the problems faced with autonomous driving.

As noted earlier, machine learning training requires high end CPU/GPU workstations or servers. These high-end computers learn by processing hundreds of thousands of images for training and inferencing. Typically, the more GPUs added to the server will increase the number of images processed, but there are other considerations as well, including the access speed to the image storage. Even local SSD and NVMe storage becomes a bottleneck, so ultra-fast parallel file systems are often required to obtain the appropriate throughput. When a fleet of vehicles collect data for machine learning training and inferencing testing, a parallel file system likely will be required in order to provide data scientists with fast enough turnaround on cycles. Data scientists tend to like to experiment, and if a 12-hour cycle can be cut down to four hours, and data can receive more churning, then the overall R&D program is likely to benefit.

Validate + drive

This is the phase of the development cycle where program requirements are validated and verified. Typically, 90% or more of the validation effort will occur in simulation verses real-world driving. In order for autonomous vehicles to be accepted by the general public, they will need to be perceived to be as safe — or safer than human drivers — and according to a recent Rand Corporation study, “to drive even 20 percent better than a human requires 11 billion miles of validation. That translates to more than 500 years of nonstop driving in the real world with a fleet of 100 cars – an impossible task.”

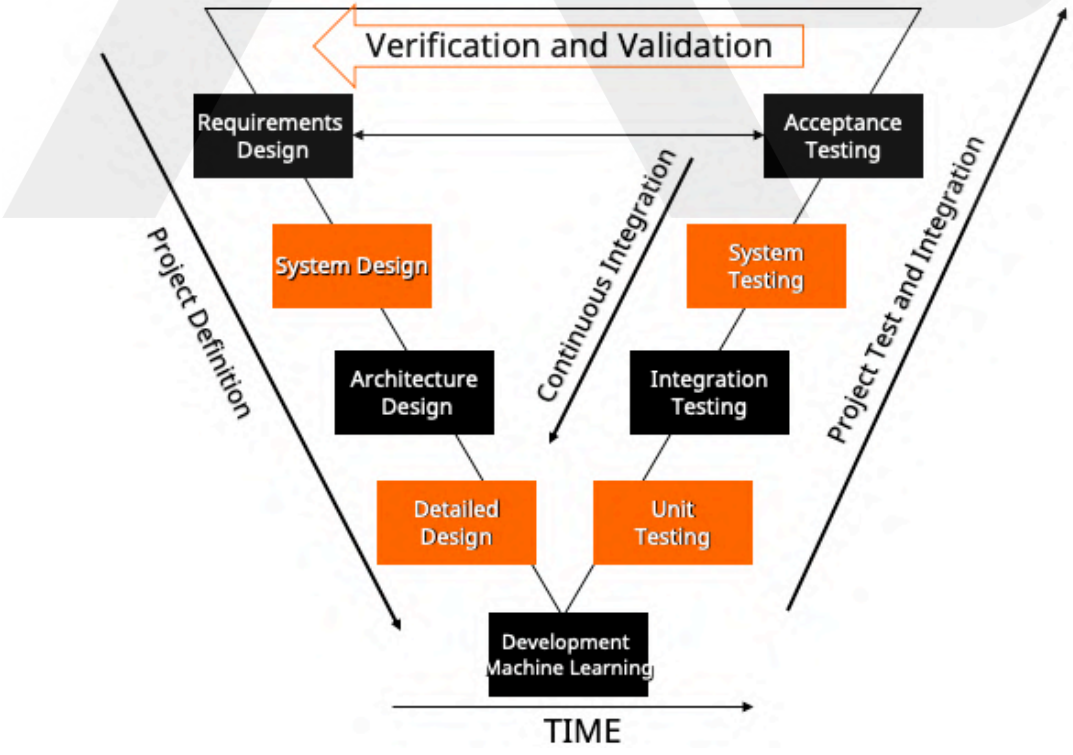
Validation and testing are sometimes interchanged terms, but testing occurs in multiple phases of the R&D cycle including unit test, system test, integration test and acceptance testing. The term validation typically applies to acceptance testing, as this is when an independent team validates and verifies that the system meets the original requirements. The V model diagram below shows where the various types of testing fit into the development cycle. The V model represents a traditional water fall development approach, but even with an agile mindset, the different types of testing still apply. The testing tools vary across the testing phases, and developers will choose tools that allow them to debug modules easier on their workstations. However, farther up the testing chain, the testing tools will focus on full autonomy systems and validating high level requirements. The key is to choose the right tool for the appropriate testing phase, and there is no one testing or simulation tool that works well for all of the testing phases.

Simulation tools used for verification and validation are typically designed so they can run thousands of test cases in parallel and faster than real-time. Quite often, these are Software in the Loop (SIL) solutions, and in more advanced R&D

programs will run in a regular cadence and as part of a continuous integration development process. The regular cadence would be tied to a regular software build schedule, so newly trained machine learning models would be continuously validated and regression tested.

An added benefit of simulation is the generation of synthetic data for machine learning training. Test cases are run in the simulation tool, and a test case is created for a condition that is hard to create in the real world, such as a broken yellow traffic light. This edge test is created, and the data from the sensors is exported from the simulation tool and fed into a machine learning algorithm in training mode. So, instead of trying to find a broken yellow stop light or creating this real-world scene, many broken yellow light test cases can be created to train machine learning models.

The final step in the validation and verification phase is real-world driving. This is autonomous vehicles operating on closed test sites and finally on public roads. Typically, check drivers are placed behind the wheel, so a human can take control during validation. Events where the operator takes control and other “interesting” activities are noted so machine learning models and software can be improved. Many R&D programs collect data from the autonomous vehicles and feed it back into simulation for continuous replay and refinement.



Conclusion

AutonomouStuff engineers have spent countless hours with technology suppliers, interviewing our most savvy customers and learning from our own development experiences to better understand the complete R&D cycle. This topic is complicated. It can be incredibly expensive and painful if not architected, from the beginning, very carefully. You can learn more by visiting https://autonomoustuff.com/data-intelligence/ or contacting AutonomouStuff's support team at the contact information provided below.

Contact information

Address: AutonomouStuff
306 Erie Ave.
Morton, IL 61550

Phone: (309) 291-0966
Fax: (309) 481-5425
Email: support@AutonomouStuff.com
Web: www.AutonomouStuff.com