Guidelines for Quality Assurance of AI-based Products and Services

2021.09 (Informal English version by machine translation)

Consortium of Quality Assurance
for Artificial-Intelligence-based products and services
(QA4AI Consortium)

Consortium of
Quality Assurance for
Artificial-Intelligence-based
Products and services
ＡＩプロダクト品質保証コンソーシアム

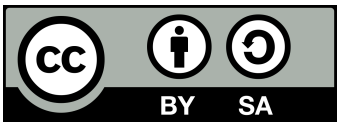
About this Document

This document is intended to provide a set of common guidelines for quality assurance of AI products. Since AI technologies such as machine learning are still in their infancy, and these guidelines are not intended to be exhaustive or complete. Therefore, each organization should use these guidelines as a guide and reflect the situation of their domain, their company, and their organization.

Neither the Consortium for Quality Assurance of AI Products, the originator of these guidelines, nor any individual or organization belonging to the Consortium shall be responsible for the quality of AI products developed or provided in accordance with these guidelines.

These guidelines do not represent the official views of the individuals belonging to the Consortium or the organizations to which they belong, or the organizations belonging to the Consortium or their superior organizations, etc.

The original version of these guidelines was written in Japanese and, for the sake of rapidity, the English version has been produced by machine translation. It may contain translation errors.



History (Original in Japanese)

May 17, 2019	2019.05 version published	first edition
Feb. 1, 2020	2020.02 version released	added chapter on AI-OCR
Aug. 1, 2020	2020.08 version released	revised
Sep. 15, 2021	2021.09 version released	revised

CAUTION

The original guidelines were written in Japanese. This document was translated into English by machine translation to provide a quick reference. The QA4AI consortium is not responsible for the quality of the translation.

Contents

1. Objective and scope

1.1 Background and purpose

Machine learning and other AI technologies have been evolving and spreading, becoming not only a source of competitiveness for various industries, but also disrupting and transforming existing industrial structures and creating new industries. As a result, the impact of products and services based on AI technology (AI products) on people's lives, society, and the economy is also increasing.

On the other hand, AI technology is still in the process of evolution, and due to its technical characteristics, it is much more difficult to understand, evaluate, explain, and control its quality than hardware, conventional software, and services. In particular, since the behavior of machine learning is inductively determined by learning data, conventional quality assurance methods for software cannot be used. The contribution of quality assurance through development process management is also small. Therefore, it is difficult to say that the quality assurance technology for AI products has been established. In other words, our lives, society, and economy are inherently at great risk of quality incidents in AI products.

At the same time, it is important to note that society's excessive expectations for the quality of AI products, ignoring the technical characteristics of AI technology, will require excessive activities for quality assurance, which will put pressure on the appropriate use, timely release, and further evolution of AI technology. We need to mitigate the quality risk of AI products and prevent excessive quality pressure, so that AI technologies can be used and evolve with confidence.

Therefore, there is an urgent need to investigate, systematize, support, apply, and research and develop quality assurance technologies for AI products in order to ensure a safe and secure life, society, and economy. At the same time, it is necessary to promote awareness-raising activities so that society can have an appropriate understanding of the quality of AI products based on their technical characteristics.

Therefore, we are issuing guidelines for quality assurance of AI products. These guidelines provide a common guideline for quality assurance of AI products to prevent excessive expectations of AI technology in each organization, and to ensure appropriate use and timely release.

Because AI technologies, such as machine learning, are evolving rapidly, these guidelines will be updated periodically on an annual basis. Consideration should be given to this when developing standard documentation and assessing the degree of conformance in each organization and industry.

1.2 Quality Assurance Issues for AI Products and Scope of this Guideline

AI technologies that form the basis of AI products can be broadly divided into rule-based technologies that can be developed deductively and machine learning technologies that can be developed inductively. The former can be assured by traditional quality assurance techniques, while the latter is difficult to be assured by traditional quality assurance techniques.

Deductive development is a way of development in which the internal design and implementation can be explicitly related to the defined specifications, and various verifications can be performed based on them. The relationship between the development process and product quality is relatively clear. Therefore, the experience to improve the quality is accumulated as knowledge, and the means of quality

assurance focusing on internal design and implementation review, metrics, and processes are effective.

On the other hand, in inductive development, internal design and implementation cannot be explicitly related to the defined specifications. In other words, it is extremely difficult to review whether the internal design and implementation are good or bad, and it is also impossible to evaluate the quality by statistical metrics. Likewise, the relationship between the development process and product quality is likely to be unclear, so the means of process quality assurance are extremely limited. For example, it is difficult to estimate the number of remaining bugs, and process audits do not have the desired effect.

Therefore, in inductive development, we often proceed with a style that should be called exploratory development. This is a style of small-scale and iterative development, such as learning and building products little by little, and demonstrating that quality has been improved and ensured through testing, trial runs, and actual operations.

There are two types of machine learning technologies: those that can assume linearity and distribution, and those that cannot. Many of the technologies currently used in AI products are the latter, non-linear technologies. In addition, in neural networks, many neurons have very complex structures.

Quality assurance is achieved by predicting the validity of behavior under all conditions in the future, but it is practically impossible to consider all conditions from the standpoint of cost and delivery time, and future predictions are unknown in principle. However, from the standpoint of cost and delivery, it is impossible to consider exhaustive conditions, and future predictions are unknown in principle. On the other hand, nonlinear techniques and distributional techniques have been used.

On the other hand, when using nonlinear technology or technology with a very complex structure for which no distribution can be assumed, it exhibits a property called CACE (Changing Anything Changes Everything: the property that even a small change affects the entire system), and the quality is not based on linearity, distribution, or divide and conquer. It is not based on linearity, distribution, or divide-and-conquer, but on the need to guarantee quality every time a learning or change is made. FEET (Frequent, Entire and Exhaustive Testing) is required to test the entire component frequently under all conditions.

At the same time, it is extremely difficult to explain and understand the causal relationship between misjudgments, even when they are corrected by a group of learnings, because the structure is nonlinear and extremely complex with no distributional assumptions. Such explanatory techniques are now being actively researched as eXplainable AI (XAI).

In the quality assurance of AI products, the quality of the model, which is the core of the AI component, and the quality of the data that determines the model are the most important. The quality of the model and the quality of the data are widely discussed as inherent techniques in the fields of statistics and machine learning. Quality assurance can follow the same approach, but it is also necessary to consider practical aspects such as large dimensional data, real data that can be obtained, and online learning. It is also important to understand that in mission-critical domains, the accuracy of the model is basically not 100%.

An AI product development organization has two aspects: data science and software development. Development organizations that are strongly positioned as data science organizations may trivialize the accuracy of the model as quality. Development organizations that are strongly positioned as software development organizations may blindly believe that processes and metrics are the key to quality assurance. When the former type of organization discusses quality assurance of AI products, it is necessary to consider how to assure the quality of AI products as a whole system. For example, it is necessary to understand the value of the system, evaluate the criticality of possible quality incidents, and estimate the frequency of occurrence of events that may cause quality incidents, etc. It is necessary to recognize that the concept of quality assurance in deductive development is useful.

When the latter organization discusses the quality assurance of AI products, it is necessary to take the

perspective of comparing organizations that have achieved high quality in the first place with those that have not. In deductive development, there are many organizations that have low quality even though they comply with processes and metrics. On the other hand, organizations where technically skilled developers are strongly convinced and share that conviction strongly in their teams and organizations, as well as with customers, generally have high quality. It is not because they adhere to processes and metrics that they have strong conviction empathy. It is because the development team strongly shares the sense of conviction, and as a result, the development process is technically necessary and sufficient, and the metrics are achieved. Such organizations tend to develop in an exploratory manner, which makes it easier to strengthen the empathy of conviction.

Another major concern in quality assurance of AI products is customers who have little understanding of the characteristics of AI products. Regardless of whether the development is deductive or inductive, if the expectations are high to begin with, the quality assurance needs to be more robust. Furthermore, if the customer does not have a good understanding of the characteristics of AI products, there is a risk that quality assurance will be difficult.

Customers who do not understand the characteristics of AI products may think that the AI product and the development organization will always do everything perfectly without them having to do anything in particular. Of course, this is a misconception. These customers are not aware of the quality and quantity of data, do not allow exploratory development and do not give the necessary authority, and reject the need for change in the customer's organization. They may demand 100 percent accuracy, reject nonlinear behavior, and demand reasonable and detailed explanations of the model's behavior. Properly controlling these customer understandings is important for quality assurance of AI products.

In this guideline, we will consider the above issues in quality assurance of AI products. First, a framework of basic concepts and considerations is described in Chapter 2, which presents the five axes to be considered in the construction and evaluation of AI product quality assurance: Data Integrity, Model Robustness, System Quality, Process Agility, and Customer Expectation. Expectation that should be considered in building and evaluating the quality assurance of AI products, and discuss the balance and margin between them. Next, we organize a catalog of technologies for promoting quality assurance of AI products. In addition, we will discuss the technological trends related to explainability and interpretability in machine learning, which have attracted particular attention in recent years. Then, based on these trends, we will illustrate individual guidelines for four domains: content generation systems, smart speakers, industrial processes, and automated driving.

Since AI technologies, such as machine learning, are evolving at a remarkably fast pace, these guidelines will be updated periodically on an annual basis. This should be taken into account when developing standard documents and assessing the degree of conformance in each organization or industry.

Similarly, since AI technology is still in its infancy, these guidelines are not intended to be exhaustive or complete. Therefore, each organization should use these guidelines as a guideline, carefully consider and reflect the situation of its domain, itself, and its own organization, and utilize them under its own responsibility.

When discussing quality assurance of AI products, the question of how quality assurance should be stipulated in contracts cannot be avoided. This guideline does not include contractual issues in its scope at this time, but the following is a brief perspective for consideration.

The expression "quality assurance" tends to be taken in the context that the vendor is responsible for the quality of the AI product to be developed, but the content of quality assurance in these guidelines is not solely the responsibility of the vendor by nature, and the content of responsibility varies. For example, there are many cases where the vendor should be held responsible (the vendor should guarantee non-infringement of copyright) if the AI product infringes on a third party's copyright. On the other

hand, in many cases, users should be responsible for providing data used for learning and for providing such data in a state where it can be legally used. In addition, inevitable risks, such as the probabilistic behavior of AI products, should be considered not from the perspective of which party bears the obligation, but from the perspective of whether the risks have been properly explained and whether appropriate risk sharing has been defined.

In general, in system development, judicial precedents have pointed out that the vendor may be obligated to project management and the user may be obligated to cooperation, and in the development of AI products, both the user and the vendor may have obligations, and the nature of those obligations may vary. In the development of AI products, both users and vendors may have obligations, and the nature of these obligations may vary. It is expected that discussions on this point will be deepened from the viewpoint of who should be responsible for each item and how it can be reflected in the contract, using the checklist in this guideline as a guide.

In addition, if the quality of AI products in a contract is considered to be incompatible with the contents of the contract, or if the vendor is considered to have breached the duty of care in the development process in a quasi-delegated contract, the vendor's liability to the user may become an issue. However, if there is no provision for quality in the contract, it is not clear what level of quality the vendor is liable for if it falls below. As a result, problems may occur due to the lack of clarity, which may lead to an undesirable situation for both users and vendors. In order to eliminate such risks, it is necessary to have an in-depth discussion on how the quality of AI products should be defined in contracts.

2. Quality Assurance Framework for AI Products

2.1 Basic approach to quality assurance of AI products

2.1.1 Axis to be considered in quality assurance of AI products

First of all, in order to develop a quality AI product, the data must be in order. In other words, it is important to secure appropriate and sufficient data in terms of both quality and quantity, and data for training and data for validation must be independent. The axis that should be considered from this perspective is called Data Integrity in this guideline.

Similarly, in order to develop a high-quality AI product, the model must be well-defined. In other words, a model with high accuracy and robustness is important. It is also necessary to properly deal with degrading in training and other processes. The axis that should be considered from this perspective is called Model Robustness in this guideline. In this guideline, the term "model" refers to a learned model, or an instance of a model, and the type of model is called an "algorithm. This guideline covers not only models based on a single algorithm, but also models based on a combination of algorithms. In the case of AI products that do not use any machine learning components, quality assurance is not explicitly addressed in this guideline, as quality assurance for deductive development is also valid. However, quality assurance is of course necessary for non-machine learning AI products as well, and the descriptions in this guideline that are not specific to machine learning components may be useful as well.

Also, in order to develop quality AI products, the system as a whole must have high value and be able to cope even if something happens. In other words, it is important to guarantee that the quality of the entire AI product is ensured. The axis that should be considered from this perspective is called System Quality in this guideline. However, this guideline does not intend to define a superordinate or subordinate relationship between similar concepts such as Quality, Reliability, Dependability, Safety, and Security in a narrow or broad sense, but uses the concepts of Quality and Quality as a set of such concepts. Instead, the notion of quality is used as a set of such concepts. Therefore, depending on the domain, it may be useful to replace Safety, Security, Dependability, and so on.

In order to develop quality AI products, it is necessary for developers and development teams who share the same sense of conviction to proceed with exploratory development flexibly using automated development environments. In other words, it is important for the process to be agile. The axis that should be considered from this perspective is called Process Agility in this guideline.

In order to develop quality AI products, it is important to have a good relationship with customers. In other words, it is important to have high customer expectations, both good and bad. If the customer expectations are high in a good sense, quality assurance needs to be done well, and if the customer expectations are high in a bad sense, we have to deal with the risk of difficulty in quality assurance due to customers who have little understanding of the characteristics of AI products. The axis that should be considered from this perspective is called Customer Expectation in this guideline.

In the next section, we will present the considerations for evaluating and building quality assurance for AI products according to these five axes. However, since the technology in the field of machine learning is still in the development stage, the considerations presented may not be exhaustive. Each

organization using this guideline will need to add considerations as appropriate while utilizing these five axes.

2.1.2 Data Integrity

Data Integrity considers the securing of adequate and sufficient data in terms of quality and quantity, and whether data for training and data for validation are independent. Data Integrity has been widely discussed as an inherent technology in the fields of statistics and machine learning, so it should be followed in quality assurance.

First, the amount of data must be sufficient and the cost must be appropriate. However, it must be a meaningful amount. For example, it is possible to change the brightness or color of image data by performing operations or transformations to make it different, but if the generalization performance is degraded as a result, it is not a meaningful increase in quantity.

In terms of data quality, it is necessary to consider whether the statistical properties of the sample are met. It is necessary to consider whether the sample belongs to the desired population, whether it is actual data or artificially created data, whether it contains unnecessary data, noise, or data from different populations, etc. It is also necessary to consider whether there is any bias, bias, or contamination, and whether the source of the bias should be the one we think it is.

Since the data handled by AI products are often high-dimensional and complex, such as images, it is necessary to consider whether the sample contains the necessary elements appropriately, and whether it is too complex or too simple. The properties of the data, such as multicollinearity, also need to be considered. In the case of supervised learning, we also need to consider the validity of the labels.

We also need to consider outliers and missing values. It is necessary to consider whether each data is common sense, whether outliers are really outliers and have no meaning, whether there is no meaning in the reason or background of the missing data, and whether the handling of outliers and missing values is appropriate.

Since AI products are repeatedly trained and validated, training data and validation data should be independent in many cases. In this case, it is necessary to consider the mechanism to ensure independence and confidentiality. In the case of artificially created data, the quality of the data generation program must also be guaranteed. Similarly, the quality of the learning program and the learning process must also be guaranteed. Otherwise, the meaning of the data will be compromised.

In addition, when conducting online learning, it is necessary to consider what kind of data will be given, what kind of learning may be conducted, and what kind of impact this may have.

In addition to the above, when using actual data, it is necessary to consider the legal and ethical aspects. Specifically, it is necessary to consider whether there are any restrictions on the use of the data from the perspective of contractual restrictions, restrictions on the rights of third parties, restrictions under laws and regulations, and restrictions due to issues such as ethics and privacy.

In other words, first of all, the use of data may be restricted by contractual obligations, such as confidentiality clauses or conditions of use of the data.

In addition, the use of data may be restricted due to the fact that the rights of third parties, such as copyrights, extend to the data (with regard to copyrights, it is necessary to consider that even works of third parties may be used for the purpose of information analysis such as learning (Article 30-4 of the Copyright Act). (Article 30-4 of the Copyright Act). In addition, the data may contain personal information.

In addition, if personal information is included in the data, it must be handled in accordance with the Personal Information Protection Law. Compliance with laws and regulations such as the Unfair

Competition Prevention Law, which protects trade secrets and data provided on a limited basis, is also required.

In addition, there may be restrictions on the use of data due to privacy considerations, handling of ethical issues, and so on.

It is necessary to take appropriate measures to address these points when using data.

2.1.3 Model Robustness

Model Robustness considers the accuracy and robustness of the model, degrading, etc. Like Data Integrity, Model Robustness has been widely discussed as an inherent technology in the fields of statistics and machine learning. We can follow the same approach for quality assurance.

First of all, it is necessary to sufficiently consider accuracy, such as correct answer rate, fit rate, recall rate, and F-value, as well as generalization performance. It is also necessary to take into account the AUROC (Area Under Receiver Operating Characteristic: AUC in the ROC curve), which is an indicator of the goodness of the model.

Accuracy and generalization performance need to be considered at an appropriate frequency for each training. In doing so, it is also necessary to consider whether the learning has progressed appropriately and whether the model has not fallen into a local optimum.

As the learning progresses, it is necessary to consider whether the algorithm is appropriate, whether the hyperparameters are appropriate, and so on.

In terms of model validation, it is necessary to consider whether the model has been sufficiently cross-validated, whether the model is robust to noise, and whether the model has been validated with sufficiently diverse data. Not only mathematical diversity, but also semantic diversity, social and cultural diversity should be considered.

We also need to consider how to deal with degrading as the learning progresses. Degrading refers to a phenomenon in which data that were correctly discriminated before a certain learning process are misclassified after the learning process. It is necessary to consider whether the degrading is acceptable or not, and whether the range of influence of the degrading is properly understood. When dealing with an unacceptable degrade by learning, it is necessary to pay sufficient attention to the degradation of generalization performance. In order to properly consider degrading, training must be reproducible. In addition, it is necessary to examine whether the behavior during operation is consistent with the behavior during training.

It is also necessary to consider whether the model will become obsolete or not during the learning process and during the mid-to-long-term operation process. It is also necessary to consider whether or not the quality of prediction for actual data will deteriorate. Whether the obsolescence or degradation is caused by the data, the hyperparameters or algorithms, the system design, or the service concept should also be considered and addressed.

In addition, when conducting such a study, the target metrics may not be directly measurable like the metrics in actual services such as customer satisfaction. In such a case, metrics that can be measured are used as substitute characteristics for the target values, but it is necessary to consider whether the relationship between the target metrics and the measured metrics is appropriate.

2.1.4 System Quality

In System Quality, the quality of the entire AI product is taken into account, and by considering the AI component as a special component, there is a possibility that the conventional quality assurance

know-how of deductive development can be utilized.

First, it is necessary to consider whether the system as a whole is providing value appropriately. What the value of the system means depends on the system, domain, and business model, but in the case of AI products, there are still many cases of "just try it", so it is necessary to iteratively examine how we perceive the value of AI products.

Since AI products as a whole are a mixture of deductive and inductive development, it is possible to divide and conquer the former, but difficult to divide and conquer the latter. It is important to identify the difference.

In quality assurance, fatal quality degradation, its effects, and the entire event including both are generally referred to as quality incidents, and it is necessary to consider whether the fatalities of quality incidents that may occur in AI products can be controlled to an acceptable level. From the sound of the word "accident," it may seem that only harm to the body or life is considered, but depending on the domain, quality accidents can be considered in various ways, such as economic damage, impact on society and the environment, discomfort, unattractiveness, lack of meaning, and unethicality.

As for quality incidents of AI products, it is necessary to consider not only functional quality incidents such as misclassification, but also whether the behavior of the entire system will deteriorate such as performance degradation and usability deterioration.

When considering quality incidents, it is necessary to fully consider the triggering events. The triggering events need to be examined comprehensively, and the frequency of occurrence of each triggering event also needs to be examined. Trigger events can occur outside the AI product, such as a pedestrian suddenly jumping out of the way, or they can occur inside the AI product, such as a bug. Sometimes, the frequency of occurrence can be controlled by controlling the usage environment. In this case, environmental controllability can be classified by system, environment, user, usage, domain, business model, etc. For example, suppose a certain function is a trigger event that causes quality incidents. There are cases where the developer can intentionally restrict the use of users who cause the triggering event by means of warning messages, disclaimers, etc. (intentional), cases where the developer cannot intentionally restrict the use of users who cause the triggering event because the triggering event occurs accidentally (accidental), and cases where the developer cannot ignore the triggering event no matter how small the probability of occurrence is because a security attack is expected (attack). The three types of cases must be distinguished because they differ in the degree and method of quality assurance.

It is also necessary to consider the degree to which a quality incident can be reached from a trigger event and the degree to which damage can be reduced. For example, it is necessary to consider the existence and number of protective mechanisms, safety functions, and attack resistance, and whether they are good or bad. However, it should be noted that overly complex protective mechanisms may have adverse effects on the overall quality of the system, such as an increase in the number of trigger events, an increase in the degree to which quality incidents are reached, and an increase in the damage caused by quality incidents. For the system as a whole, there are cases where the degree of arrival and damage of quality incidents can be reduced by increasing avoidability and controllability, and self-healing properties can be provided, so these points must also be considered.

In the case of AI products, it is necessary to consider the structure of AI components and non-AI components. AI contribution also includes considerations such as whether changes in both AI and non-AI components can be reflected quickly and appropriately, and whether the impact of failures can be kept low enough.

It is also important to ensure that stakeholders have confidence that the quality assurance activities can truly guarantee quality. Ensuring such assurance, accountability, and conviction is not so easy in the case of AI products. Ensuring them through processes and metrics, as in deductive development, is

meaningless. It is not practical to manually create and maintain documentation for exploratory development and FEET. The technology of eXplainableAI is still in its infancy. Therefore, to ensure assurance, explainability, and acceptability, it is extremely important to ensure the empathy of acceptance among quality assurance engineers and departments, just like the empathy of acceptance among developers and teams. Quality assurance personnel and organizations themselves must always be convinced of the meaning of their quality assurance activities, how they technically contribute to quality improvement, how they increase the empathy of conviction among developers and teams, and how they make efforts to prevent them from thinking that their work is wasteful or meaningless. Quality assurance people and organizations must always be convinced that they are doing their best to increase the number of people doing the work and to prevent them from thinking that the work is wasteful or pointless. Quality assurance personnel and organizations must be aware that they are engineers, not managers.

It is also necessary to consider whether the AI products infringe on the copyrights, patents, or other intellectual property rights of third parties, whether the use of open-source software violates the conditions of use imposed on the software, and whether the use of AI products violates laws and regulations.

When AI is embedded in a device, it is also important to share the responsibility between the manufacturer of the device and the developer of the AI in the event of a problem with the quality of the device.

In other words, when AI is embedded in a device, the quality of the AI product may appear as a product liability of the device (a typical example is self-driving cars, which are also covered in this guideline). This is because although AI models themselves are not tangible and thus do not fall under the category of "movable property" (Article 2, Paragraph 1 of the Product Liability Law) subject to product liability, when embedded in a device, the quality of the AI model can be an issue as a "defect" (Article 2 of the Product Liability Law) in the device. Basically, however, the "manufacturer" (Article 3(1) of the Product Liability Law) who bears product liability is considered to be the manufacturer of the device itself and not the developer of the AI model. Therefore, in many cases, it is the manufacturer of the device, not the developer of the AI model, who is directly claimed by the victim for damages when the device is defective. However, if the manufacturer of the device is held liable by the victim and believes that the cause of the defect is the AI model, the manufacturer may pursue the developer of the AI for liability under the development contract. In this way, it should be noted that it may be desirable in some cases to stipulate the division of responsibility in the contract, because any damage caused by a defect in the equipment may ultimately become a question of who should bear the damage, the manufacturer of the equipment or the developer of the AI.

2.1.5 Process Agility

Process Agility considers the agility of the process: in order to guarantee the quality of AI products, developers and development teams who share a sense of conviction need to proceed with exploratory development flexibly, making full use of the automated development environment. Therefore, it is necessary to consider whether the development is flexible, whether it is sufficiently automated, and whether the developers and development team share a sense of conviction.

In order to develop AI products flexibly, the speed and scalability of data collection need to be sufficient. At the same time, iterative development with sufficiently short iterative units and sufficiently short cycles for improving the quality of models and systems is necessary. Continuous feedback on operational status must also be frequent.

It is necessary to consider whether the model or system is likely to get better as a result. Consideration should be given not only to the progress of learning, but also to the ability to rapidly add new features

and rapidly improve the model. In order to do this, we need to have the means, environment, and mechanisms in place to quickly debug training and inference.

AI products may have unexpected quality incidents due to their non-linearity, and in such cases, release rollback needs to be simple and quick. Similarly, it is necessary to consider whether the frequency and degree of staged releases and canary releases are appropriate, and whether the entire model and system are evaluated and tested at the time of each release.

In order to perform exploratory development and FEET during training and release, it is necessary to automate development, exploration, validation, and release. In doing so, it is necessary to have appropriate configuration management for data, model, environment, source code, and output.

Since the development and quality assurance technologies for AI products are still in their infancy, developers and development teams contribute to the quality of the products. It is extremely important that the developers and the development team are technically satisfied and empathetic with the behavior of the product and the development process, and that the customers are also satisfied and empathetic. We need to be convinced and empathetic about how we are making things, whether we are developing and exploring in a reasonable way, what we should do if something happens, how far we should go back, what kind of risks there are, whether we generally know how to deal with such risks, and to what extent we know what we do not know. We need to empathize with them. However, due to the unique nature of AI products, it is extremely difficult to perfectly anticipate all risks and how to deal with them, so an empirical sense of balance between what we know and what we don't know is also necessary.

The development team must also be equipped with the right people with the right capabilities to increase the confidence level of the conviction. Properly skilled personnel are needed for each of the expertise in data science and machine learning, software development expertise, and domain technology. In doing so, it is necessary to evaluate them based on their practical experience as well, rather than relying only on qualifications and certification systems that can be obtained through classroom lectures.

In the development of AI products, it is necessary for developers and development teams to reflect various "learnings" (i.e., experience, trial and error) in the acquisition of insights and improvement of technology as they proceed with exploratory development. In addition to technical reflection measures, human reflection measures such as looking back are also important.

2.1.6 Customer Expectation

Customer Expectation considers the relationship with a good customer; whether the customer's expectations are high or not is important to ensure the quality of AI products. In order to assure the quality of AI products, it is important to know whether the customer's expectations are high or low. If the customer's expectations are high in a positive sense, it is necessary to do quality assurance well, and if the customer's expectations are high in a negative sense, it is necessary to deal with the risk of difficulty in quality assurance due to customers who do not understand the characteristics of AI products.

First of all, when customer expectations are high in a positive sense, quality assurance needs to be done well. AI products that can cause damage to the customer's body or property in the event of a quality accident, such as automated driving or financial transactions, have higher expectations than AI products that do not bother the customer in the event of a quality accident, such as free hobby-type products. This is true regardless of whether the development is inductive or deductive.

However, if the customer's expectations are high in a negative sense, i.e., if the customer has little understanding of the characteristics of AI products, quality assurance may be at risk of becoming difficult. Such customers may think that the AI product and the development organization will always do everything perfectly without them having to do anything in particular. Of course, this is a misconcep-

tion. Quality assurance engineers, teams, and organizations, along with development and sales, must control customer expectations appropriately by working to deepen the customer's understanding of the AI product.

It has been said that AI products work probabilistically. Since AI products are computer programs, if the same input is given to an AI product in the exact same state, it will return the exact same output. In this sense, the AI product works deterministically. In this sense, the AI product works deterministically, but it seems to work probabilistically for the entire set of possible inputs, because the accuracy is not 100%. And because it behaves in a non-linear manner, it may produce different outputs even though the inputs seem to be the same to humans, and its behavior may change unexpectedly as it learns, so it seems to behave in a stochastic manner. In addition, there are algorithms that can produce different models even with the same data, the same algorithm, and the same hyperparameters (except for the initial values), so they seem to work stochastically. Some customers have a habit of demanding "no defects" in deductive development, and may not tolerate such stochastic behavior of AI products. This can lead to over-learning, which can reduce quality, or create unnecessary work and slow down the development process. The same is true if the customer does not understand or tolerate the risks and side effects.

Since AI products involve exploration, it is rare to conduct development at the production level right away. Therefore, Proof of Concept (PoC) and beta releases are often conducted, and quality assurance must be conducted accordingly. However, if the customer does not understand the development stage such as PoC or beta release, they may point out risks that may cause problems in the production stage and ask for solutions even in the PoC stage. This may also lead to a drop in quality and stagnation of development.

Some customers may not be aware of the quality and quantity of their data. Customers do not always understand what kind of data they have and how much of it they have, and how they can obtain it from outside the company. It is necessary to make them fully aware of this before development or contracting.

Whether it is an AI product or deductive development, it is necessary to clarify what the purpose is, what data is used, with what accuracy, and what kind of output is desired. Sometimes, customers demand "human-like" accuracy. In such cases, it is necessary to examine and confirm whether the stated and agreed upon accuracy target is the same as what the customer implicitly intends. For example, if the explicit goal is to avoid an obstacle, the overall behavior of the customer may be such that the human does not avoid the obstacle by grazing the tip of its nose. In addition, AI products are basically unable to judge ethical issues such as trolley problems, which are difficult to judge even with human experience, sense, formulas, and rules. When a customer requests "human-like," it is necessary to discuss and agree on the meaning of the request, or have the customer understand the process of clarification through PoC exploration.

Depending on the domain, it may be necessary to consider whether the use of AI products infringes on the privacy of third parties, or whether there are ethical issues. If a similar system or mechanism does not yet exist, social acceptance may be necessary; since AI products may play an advanced role, customers may take these points lightly.

Some customers may not understand the virtual black-box nature of stochastic behavior and the complexity of the structure of machine learning models, and may demand reasonable and detailed explanations, extrapolations, and predictions of misjudgments, defects, and risks that occurred during development. This can lead to waterworks and stalled development.

Customers with such a poor understanding of the characteristics of AI products often lack a culture, atmosphere, and work style that empathizes conviction between the customer and the developers and team. Similarly, the customer's staff and the team often have little authority to make decisions and

have a narrow scope. Therefore, it is necessary for quality assurance to make efforts to create a culture, atmosphere, and work style that empathizes with the sense of conviction, and to create an appropriate authority and scope for decision making. You may think that this is outside the scope of work for quality assurance, which is used to deductive development. However, AI product quality assurance engineers and teams must go beyond the diminutive scope of deductive development and do everything necessary to assure quality while transcending organizational boundaries and empathizing with all stakeholders.

2.2 Quality Assurance Checklist for AI Products by Classification Axis

2.2.1 Data Integrity

(a) Sufficiency of the amount of training data
- (a.i) Is the amount of data sufficient from the statistical point of view and the application assumptions of the assumed learning method?
- (a.ii) Is there a sufficient amount of data for the assumed requirements and application environment, even if there are rare situations or biases in the classification classes?
- (a.iii) If the amount of data is small, can it be supplemented by "bulking" (e.g., artificial data generation)?

(b) Validity of the training data
- (b.i) Is the data appropriate in terms of semantics for the assumed requirements and application environment?
- (b.ii) Does the data contain any data that does not match the assumptions of the requirements/application environment?
- (b.iii) Can it be said that artificially created/processed data adequately represent the required/applied environment?
- (b.iv) Is it appropriate from the perspective of cost-effectiveness of data collection, etc.?

(c) Suitability of the training data requirements
- (c.i) Does the data meet the requirements of the stakeholders?
- (c.ii) Does the data meet the constraints on the data, such as invariance and consistency conditions, impartiality of the decisions to be learned, and availability of personal information?

(d) Appropriateness of the training data
- (d.i) Has the appropriateness of the data been confirmed by examining potential bias or contamination in terms of its impact on various stakeholders and society?

(e) Complexity of the training data
- (e.i) Is the data complex enough to contain more information or trends than necessary for the inference function to be trained?
- (e.ii) Is the data oversimplified and does it not contain necessary information?

(f) Consider the nature of the training data.
- (f.i) Are the properties of the data (e.g., multicollinearity) that are assumed to be applicable to the assumed learning method properly taken into account?

(g) Validity of the training data range.
- (g.i) Are the values contained in the data realistic and reasonable in light of the knowledge

of the target domain?
(g.ii) Have you confirmed that the values that you have determined to be outliers and missing values are not truly realistic and should be removed? Was the preprocessing to remove the data appropriate?
(h) Legal relevance of training data
(h.i) Is the use of the data not restricted by contractual or third-party intellectual property rights, does the use of the data raise any legal or ethical issues, or requires consideration of privacy or other concerns?
(i) Validity of the data for validation
(i.i) Are the training data and validation data independent?
(j) Consideration of the impact of online learning
(j.i) Are appropriate operational mechanisms and systems in place to monitor, control, restrict and verify data that is incrementally added, replaced or deleted?
(k) Validity of the data processing program
(k.i) Are the characteristics of the algorithms that perform preprocessing, creation, processing, etc. on the data, as well as their libraries and the programs that call them, defective or misused, resulting in loss of data adequacy?

2.2.2 Model Robustness

(a) Sufficiency of model accuracy
(a.i) Is the value of the evaluation metrics for inference performance, such as correct answer rate, fit rate, recall rate, and F-measure, sufficient for the requirements?
(b) Sufficiency of the generalization performance of the model.
(b.i) Is the generalization performance of the model ensured?
(c) Sufficiency of the evaluation of the
(d) model.
(d.i) Have appropriate indicators of model goodness other than accuracy (such as AUROC) been selected and sufficiently evaluated?
(e) Validity of the learning process
(e.i) Did the learning process proceed properly?
(e.ii) Does the learning result fall into a local optimum?
(f) Validity of the model structure
(f.i) Did you consider whether appropriate algorithms and hyperparameters were used?
(g) Validity of the model validation
(g.i) Did you perform sufficient cross-validation?
(h) Robustness of the model
(h.i) Robustness of the model to noise.
(i) Variety of data for validation
(i.i) Was the validation performed on sufficiently diverse data, taking into account mathematical diversity, semantic diversity, social and cultural diversity, etc.?
(j) Sufficiency of validation for model updates
(j.i) When updating the model, are you aware of any changes from previous behavior and

have you verified that they are acceptable?
(j.ii) Are the automated checks sufficient, especially for automated model updates and deployment?
(k) Consideration of model obsolescence
(k.i) Do you consider the possibility that the performance, validity, and usefulness of the model may deteriorate due to changes in trends during operation, and take measures to ensure model robustness and monitoring during operation?
(l) Appropriateness of the model as a program
(l.i) Is the model inappropriate due to the characteristics of the learning algorithm, or due to defects or misuse of the library or programs that call it?

2.2.3 System Quality

(a) Appropriateness of the value provided by the system
(a.i) Is the value provided by the system as a whole appropriate and can the value provided be measured?
(a.ii) If the value is difficult to measure, is it appropriate to relate it to alternative metrics that can be measured?
(b) Impact of AI on the system
(b.i) Does the introduction or modification of AI have a negative impact on the overall system behavior, performance, or other qualities?
(c) Validity of the system evaluation unit
(c.i) Has the system been assessed as a whole and in meaningful subsystem units?
(d) Limiting the impact of the accident
(d.i) Is the severity of potential quality incidents reduced to an acceptable level?
(d.ii) Can the frequency of events that could cause quality incidents be estimated to be low?
(d.iii) Is there sufficient consideration of the frequency of events, the comprehensiveness of the events, and the controllability of the environment affecting the events?
(e) Avoidability of accident occurrence
(e.i) Is the accident consequence of the system sufficiently controlled?
(e.ii) Does the system provide sufficient safety functions and attack resistance?
(f) Suppress the impact of AI.
(f.i) Is the degree of design dependence and coupling of various elements of the system on AI kept low?
(f.ii) Can the impact of changes to other (AI's or non-AI's) systems on which the system depends be reflected quickly and appropriately?
(f.iii) Is it possible to design or change the design so that the impact of AI failures is sufficiently low?
(g) Stakeholder satisfaction
(g.i) Is it sufficiently assured, explainable and convincing to stakeholders?
(h) Legal compliance of the AI system
(h.i) Is the AI product non-infringing on the intellectual property rights of third parties and does the use of the AI product violate laws and regulations?

(i) Consideration for quality degradation of the system
  (i.i) Have you considered the possibility of degradation in system performance and other qualities associated with ongoing operation?
  (i.ii) Have you considered a mechanism for detecting system quality degradation during operation?

2.2.4 Process Agility

(a) Speed of data collection
  (a.i) Is the speed and scalability of data collection sufficient?
(b) Rapidity of development
  (b.i) Is iterative development performed with sufficiently short iterative units?
  (b.ii) Is the cycle of model/system quality improvement sufficiently short?
  (b.iii) Does it provide frequent and continuous feedback on operational status?
(c) How quickly is the problem analyzed?
  (c.i) Does the system have a mechanism for recording and retrieving the status of the problem occurs in order to analyze the cause of the problem when it occurs?
  (c.ii) Is it possible to reproduce the event based on the obtained situation when the problem occurred?
(d) Speed of recovery
  (d.i) Can release rollback be performed easily and quickly?
(e) Rapidity of improvement
  (e.i) Is there potential for improvement, such as the ability to quickly add new features or rapidly improve the model?
(f) Rapidity of release
  (f.i) Is the degree of phased release or canary release appropriate?
  (f.ii) Is the entire system and model evaluated just prior to release?
(g) Sufficiency of automation
  (g.i) Is the automation of development, exploration, validation, and release sufficient?
(h) Adequacy of configuration management
  (h.i) Is the configuration management of data, model, environment, code, output, etc. appropriate?
(i) Suitability of the development team
  (i.i) Are the developers and team sufficiently convinced and sympathetic to the technology?
  (i.ii) Does the development team have the right people with the right skills?
(j) Rapidity of technological evolution
  (j.i) Is the experience reflected in the technology?
(k) Stakeholder satisfaction
  (k.i) Are stakeholders outside the development team fully satisfied?

2.2.5 Customer Expectation

(a) Stakeholder expectations
  (a.i) Are customer expectations high?
  (a.ii) Is it "human-like" that we are aiming for?
(b) Stakeholder understanding of the technology
  (b.i) Are customers receptive to the idea of stochastic behavior?
  (b.ii) Do they not understand the risks and side effects, or do they accept them easily and neglect to take the necessary measures?
  (b.iii) Do we have a lax understanding of the quantity and quality of data?
  (b.iv) Is there a tendency to seek "rational" explanations, to extrapolate or predict, or to assign "cause" or "responsibility"? Is there a tendency to seek "rational" explanations, to "extrapolate" or "predict", or to seek "cause" or "responsibility"?
(c) Expectations for operation
  (c.i) How close to actual continuous operation are you?
(d) Necessity of standard conformance
  (d.i) Whether there are any legal or ethical issues with the use of the AI product, whether there is a need to consider the privacy of third parties, and whether the use of the AI product is socially acceptable.
(e) Relationship with Stakeholders
  (e.i) Is there a culture or atmosphere of shared conviction and a low level of work ethic?
  (e.ii) Is there less authority or scope for decision-making by the customer representative/team?

2.3 Building and evaluating quality assurance for AI products

2.3.1 Building and Evaluating with a Focus on Balance

When constructing and evaluating the quality assurance of AI products, the five axes listed in sections 2.1 and 2.2 need to be balanced. Well-balanced means a situation in which one can expect to be able to guarantee that customer expectations are met. It consists of two conditions.

The first condition is that there is no shortage of Data Integrity/Model Robustness/System Quality/Process Agility. If any one of them is missing, it cannot be said that quality has been ensured. The other condition is that Customer Expectation is appropriate for Data Integrity/Model Robustness/System Quality/Process Agility. This means that customer expectations are properly grasped, and if possible, expectations are kept to an appropriate degree, so that Data Integrity/Model Robustness/System Quality/Process Agility are able to meet expectations overall.

An imbalance in the five axes means a situation where Data Integrity/Model Robustness/System Quality/Process Agility are insufficient to meet customer expectations. In other words, either one of Data Integrity/Model Robustness/System Quality/Process Agility is insufficient, or the customer's expectations are too high.

The action to appropriately accommodate customer expectations is often called expectation control, and both short-term, individualized expectation control and medium- to long-term, social expectation control may be necessary. In the case of AI products for specific customers, such as industrial appli-

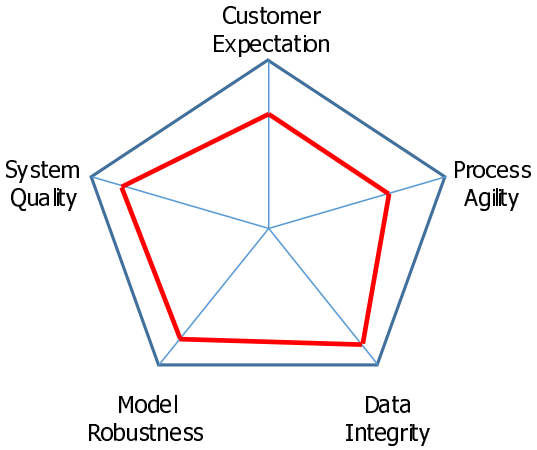


Fig. 2.1 balanced quality assurance image

cations, the former measure is important. In other words, it is necessary to identify key people in the customer's organization, make them understand the characteristics of AI products such as stochastic behavior and the value of data, and clarify and specify the true objectives and needs of the customer's organization, so that the customer does not have excessive expectations and the strategic importance of the product to the customer's business is not reduced. At the same time, it is necessary to create an atmosphere of shared conviction within the customer's organization, and to ensure that the customer's staff and teams have the appropriate degree of freedom in decision-making. In the case of AI products for general customers, the latter measures may include UI improvements, promotion of understanding through videos, and public opinion building through information dissemination in collaboration with experts, local governments, and government agencies, as well as dialogue with customers through SNS, media, and events.

Figure 2.1 shows an example of well-balanced quality assurance: the closer the pentagon plotted against the five axes is to a regular pentagon, the more intuitively balanced it is. The closer the pentagon is to a regular pentagon, the more intuitively balanced it is. Since the balance is essentially the relationship of each axis to Customer Expectation, we can also consider a notation that shows the relative relationship between the four axes and the circle representing Customer Expectation, as in Figure 2.3.

This guideline does not present the value or linearity of each axis or the relative relationship between axes. In other words, the number of checks in the 2.2 clause does not directly represent the balance. Therefore, this balance needs to be fully discussed and shared among the three parties (development, quality assurance, and customers) for each axis.

2.3.2 Construction and evaluation focusing on the development stage

When building and evaluating the quality assurance of AI products, it is necessary to change according to the development stage: the PoC stage may not require such a strong quality assurance, while the beta release stage and continuous operation stage will require a strong quality assurance. Regardless of the stage of development, however, a balance needs to be maintained. Figure 2.4 shows an image of how the size of quality assurance changes with the development stage.

This guideline does not provide the recommended values for each axis for each development stage.

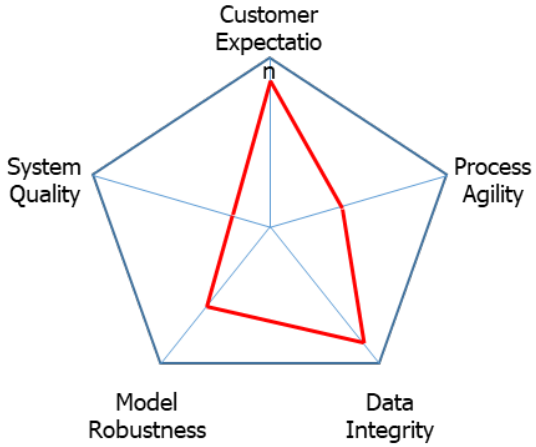


Fig. 2.2　Images of unbalanced quality assurance

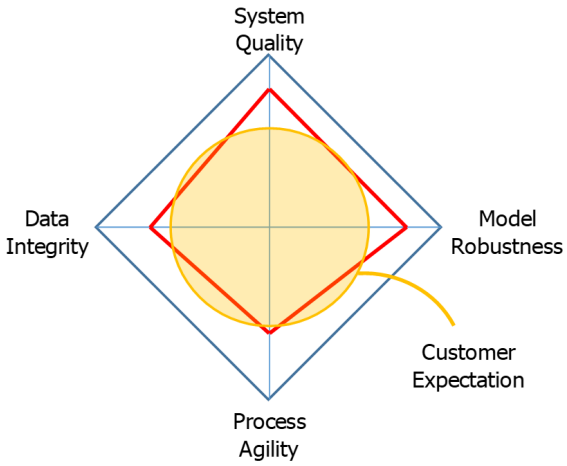


Fig. 2.3　Imaging the balance of quality assurance with squares and circles

Therefore, it is necessary for the three parties (development, quality assurance, and customers) to fully discuss and share a sense of conviction regarding each axis in order to determine to what extent quality assurance should be performed at each development stage.

2.3.3　Leftovers and Excess Quality

As we gain more experience in AI product development and quality assurance, a situation will arise where Data Integrity/Model Robustness/System Quality/Process Agility exceed customer expectations. For example, the situation shown in the orange area surrounded by the red line in Figure 2.5.

This situation can be easily misunderstood, but it is appropriate to think of it as "spare capacity". In other words, it should be interpreted as the customer’s expectations are raised when the development stage is advanced or the scope of operation is expanded in the future, so activities to prepare for that

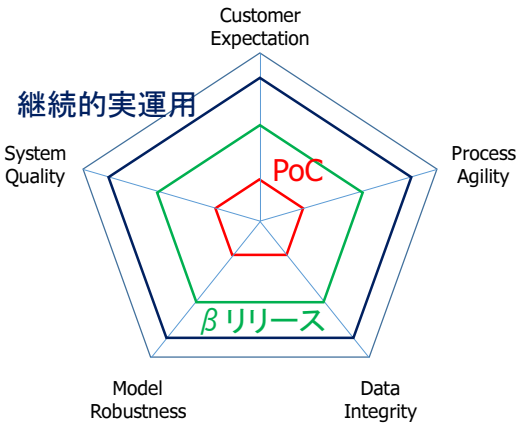


Fig. 2.4　Images of varying the size of the quality assurance according to the development stage

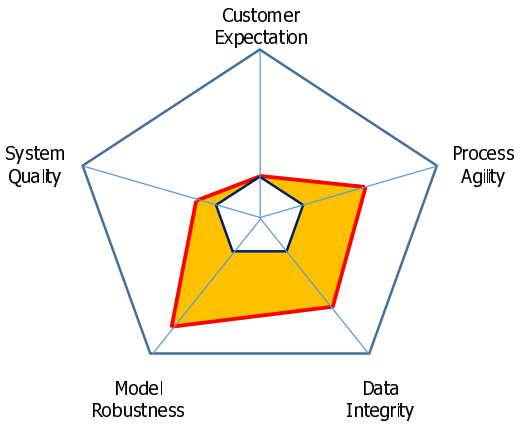


Fig. 2.5　Images of "extra capacity" for quality assurance

time are conducted in advance. In addition, having extra capacity in one project can serve as a basis for quality improvement in other projects. Therefore, it is more likely to be sustainable and overall optimal to make strategic technology investments in the direction of proactively increasing the excess capacity.

This situation should not be referred to as "excessive quality. Rather, it refers to a situation where the project or QA team loses sight of the customer's expectations, the direction of the business, the status of their own development, and their own technical capabilities, and just wastes more time without knowing what they are doing. If the number of wasted steps is increasing to achieve high quality, it is likely that the number of steps cannot be reduced due to low Process Agility or low formal meetings and documentation. Also, the accuracy and generalization performance may be too low due to low Data Integrity and Model Robustness. Or, in order to achieve System Quality, the technical essence may be neglected, and holistic and comprehensive but ineffective process measures may be promoted. In such cases, we should not wrongly label them as excessive quality, but should make appropriate investments to improve them. Otherwise, the wrong label will cause a large backlash against activities to increase the excess capacity, which will inhibit investment, resulting in a decline in quality and stagnation of development.

3. Technical Catalog

Machine learning can be categorized into three types: **supervised learning**, **unsupervised learning**, and **reinforcement learning**. In supervised learning, a model is obtained by providing a large number of pairs of input values and output values that are correct answers to the input values as training data to obtain an output for the input. In unsupervised learning, on the other hand, regularities and decision criteria are extracted from the data without defining the correct answer. Typical examples are **clustering**, which groups data, and **association analysis**, which extracts correlation rules from data. In reinforcement learning, a system learns through training what actions to take in response to a situation, such as in building a game player or a search robot. Of these, supervised learning is particularly advanced in practical applications due to its improved performance and task clarity, and this chapter focuses on the quality of supervised learning.

In the development of systems using machine learning, by using **training data** (also called **learning data** or **teacher data**) as input and executing a program that implements the **learning algorithm** (henceforth **learning program**) This is a software component that performs identification, prediction, control, etc. This component is a representation of the input-output relationship and is called **model** (the term "model" in this chapter follows this usage). In the development of systems using machine learning, training data, learning algorithms, and their parameter options (called **hyperparameters** to distinguish them from parameters whose values are determined by training) are the objects of design, and the models that are embedded and utilized in products are obtained indirectly from them. The models that are embedded and utilized in products are obtained indirectly from them.

The behavior of complex models obtained from machine learning, especially deep learning, is not written down deductively (based on general rules) by the programmer, but inductively (based on individual cases) from the training data, and is therefore a black box with the following characteristics.

- It is often impossible to clearly define the correct answer or expected value (test oracle) for all of the various input possibilities, or defining it is expensive, requiring human work.
- Functions are in principle incomplete, and even if we could define a correct answer, we cannot always seek that correct answer, and there are limits to their performance. It is difficult to estimate their performance in advance, and the boundary between what they can do and what they cannot do cannot be clearly ascertained.
- It is not possible to deductively explain how the output was obtained for each individual input.
- When the model is rebuilt with different training data, its behavior may change significantly, and it is difficult to predict the change.
- A minute change in the input (e.g., a change in the image that is not noticeable to a person) can result in a large change in the output. This is known as **Adversarial Example** 〔Goodfellow 2015〕。

These features should be taken into account in the quality of artificial intelligence systems using machine learning 〔大場 2018〕 〔石川 2018〕 〔Breck 2016〕。

3.1 Quality characteristics specific to AI products

A model obtained using machine learning cannot be 100 percent correct, even when the correct answer to the output is defined. For this reason, the most basic quality evaluation is to use known data and evaluate how many correct or near-correct answers can be obtained for them. Since the term performance is widely used in the machine learning field, we will use the term performance in this chapter as well. For example, it corresponds to accuracy in the ISO 25010 quality model (SQuaRE). Quality characteristics specific to machine learning models and systems using machine learning include **Robustness** and **Explainability/Interpretability**. In addition, depending on the target application, it may be necessary to consider quality from the perspective of reflecting cultural and social requirements, such as **fairness**of judgment.

Aspects specific to AI products are discussed below; the overall picture of quality characteristics specific to AI products is also presented in the European Ethical Guidelines [EC 2019] and in Kuwajima 2019, which discusses extensions to the ISO 25010 quality model.

3.1.1 Performance measures for supervised learning models

（Related to Model Robustness）

The basic evaluation for a supervised learning model is to evaluate its performance in terms of how well it can find correct answers to existing data. The simplest metric, **Accuracy**, is the ratio of the number of correct answers to the total number of questions. Since performance indicators such as the correct answer rate are determined relative to the data to be tested, the selection of the test data is very important. Test data that represents the assumed requirements and operational environment is necessary.

Among the models obtained by machine learning, those that deal with classification tasks that allocate input to a specific group of classes use quality indicators that have been used in the field of information retrieval. The correct answer rate may not adequately represent the classification performance if there is a bias in the class to be classified. For example, if 99.9% of the cases are normal and 0.1% are abnormal, the correct answer rate for a model that always answers "normal" (in effect, no classification) would be 99.9%.

In order to better understand the characteristics of the model, we distinguish between false positives and missing detections. First, we consider a detection task in which there are two classes to be classified: **positive** and **negative**. Of those detected (positive), those that should have been detected and those that should not have been detected (**true positive**and **false positive**), and of those not detected (negative), those that should have been detected and those that should have been detected (**true negative**and **false negative**). (**true negative**, **false negative**), respectively. These four numbers are written out as a matrix, which is called **Confusion Matrix**. The following is used as a quality indicator.

- **Conformity/Precision**. The ratio of the "number of true positives" to the "total number of true positives and false positives", indicating how many correct ones are included out of all the detected ones (fewer false positives).
- **Recall**. The ratio of the "number of true positives" to the "total number of true positives and false negatives".
- **F-Measure**。 It is expressed as $\frac{2}{\frac{1}{Fitrate}+\frac{1}{recall}}$, which indicates whether both the fit rate and the recall rate are well balanced.

There is a trade-off between the rate of fit and the rate of reproduction. The stricter the detection criteria,

the higher the rate of fit and the lower the rate of reproduction. If the detection criteria are loose, the conformance rate will be low and the reproduction rate will be high. In many cases, it is difficult to make both of them high, and it is often necessary to decide which one to prioritize based on the requirements and use cases. The **ROC Curve (Receiver Operating Characteristic Curve)**, which looks at how the trade-off between the conformance rate and the reproduction rate changes depending on the setting of the threshold used for classification, and the **AUC (Area Under Curve)**, which looks at the overall performance in this case, are also used.

If there are three or more classes to be classified, indices such as the fit rate are considered in the same way. For example, the fit rate is calculated for each class and the average is taken (called the macro average).

Among the models obtained from machine learning, for those that handle the **regression** task of predicting numerical values, a measure that captures the error between the predicted and actual values is used. Typical examples are **Root Mean Squared Error (RMSE)** and **Coefficient of Determination ($\mathbf{R^2}$)**.

The model represents the regularities and decision criteria extracted from the given training data, but during operation, values not included in the training data will be given as input. For this reason, it is necessary to pay attention to the performance (**generalization performance**) for general data not limited to the training data. Basically, it is also called evaluating **generalization error**, since the performance is degraded for data outside the training data.

Because of the importance of generalization performance, the evaluation data used for the final evaluation should be separated in advance. It should never be used for analysis of its trend, especially for training. During training, the validity of the training procedure itself is evaluated from the perspective of generalization performance by repeatedly dividing the training data into two parts, one for training and the other for evaluation, and evaluating them using different data from the one used for training, while changing the way of division (**cross-validation**).

If a model is too specialized in the regularities and criteria inherent in the training data, especially local noise and variability, and performs poorly on other input data, it is said to be in a state of **over-learning** (also called **overfitting, High Variance**). state. Conversely, if the model does not represent the necessary regularities and criteria (even within the training data), the model is said to be in a state of **unlearned**(**underfitting, High Bias**). This can be seen as a trade-off between the bias, which is the deviation between the correct answer and the prediction, and the variance, which is the degree to which the prediction is scattered. These are caused by the form of the model that represents the relationship between the outputs, the learning algorithm, and the hyperparameters. Analyze the **learning curve** to see how the performance on the training data and the performance on the evaluation data changes as the number of training data increases.

3.1.2 Evaluation of data

（Related to Data Integrity）

The quality of the training data is very important, although it is an internal quality, because it affects the quality of the models derived from them. If the output classified as the correct answer in the training data is actually incorrect or inappropriate given the original requirement, the model derived from the training data may produce the same inappropriate output. Also, when the regularity (trend, distribution) in the training data differs from the data regularity (trend, distribution) in the operation, or when a range of inputs that do not appear in the training data are given in the operation, the performance in the operation is often low. Therefore, accuracy, adequacy and sufficiency against system requirements and

assumptions of the operational environment are important as the quality of training data.

In addition, the performance indicators described in 3.1.1 are relative to the data used for evaluation. For the data used in the evaluation (training and test data), it is also important that it is appropriate and sufficient for the system requirements and assumptions of the operational environment.

In addition, it is necessary to consider general data quality characteristics such as traceability and portability defined in ISO/IEC 25012 and JIS X 25012 data quality models.

3.1.3 Robustness

（Related to Model Robustness）

The output of the model may change due to small noises in the input, such as the adversarial sample described in the beginning of this chapter. This means that the model is not robust to noise. If the model does not perform well for input values different from the training data, it is not robust to changes in the input domain. The quality property that a model achieves stable performance even when some changes are made is called **Robustness**.

3.1.4 Fairness

(Related to Dara Integrity and Model Robustness)

Fairnessrepresents the extent to which the output or behavior of a system does not exhibit an unfair bias that amounts to discrimination, prejudice, or partiality based on characteristics such as race, ethnicity, or gender [Mehrabi 2019]. Equity, by definition, involves social and cultural perspectives and is determined by the demands of various stakeholders, organizations, and societies, and requires careful consideration of what is applicable to individual systems.

It should be noted that unfairness outputs and behaviors can be embedded in the system without our awareness due to the nature of machine learning techniques. A simple case is when the training data that drives the training of the model implies unfair outputs or behaviors in the first place (e.g., when historical data containing gender discrimination is used). In the more implicit case, when using performance measures such as those listed in the 3.1.1 clause, it is easy to be satisfied with good performance for the entire given dataset. However, if we extract a specific part of the data set that contains only a small number of data (e.g., data belonging to a specific race), the performance for the corresponding input may be very poor, i.e., the performance may be unfair depending on the race, etc. Even if the system is not trained to make unfair decisions, it will result in unfair outputs and behaviors. It is necessary to define and evaluate the fairness that is important in the relevant system.

The specific definitions of fairness include demographic parity and equalized odds, where demographic parity aims to ensure that the distribution of predictive labels and predictive performance index values are consistent for sensitive attributes such as gender and race. For example, in the case of an AI that makes hiring decisions, we want to make sure that the hiring rate of males is equal to the hiring rate of females. In the latter case, we aim to make equivalent predictions for two inputs that have similar attribute values except for sensitive attributes. For example, we will confirm that a similar hiring rate is expected for two potential hires who are identical except for gender. For each evaluation criterion, there are differences in the applicable assumptions, such as whether there is bias in the data or not, and the degree of emphasis on procedural fairness versus outcome fairness. There is no social consensus on which evaluation perspective should be used when, and sufficient discussion is needed depending on the application case and its stakeholders.

3.1.5 Explainability

(Related to System Quality)

Explainability/Interpretabilityrepresents the degree to which a human using the output from a system can grasp the criteria used to obtain the output (the regularities learned by the model) [Gunning 2016]. In this case, it is the same as in the previous example. The strength of machine learning is that it can realize functions even when the requirements and behaviors cannot be explicitly written out as formal knowledge. However, explainability and interpretability are necessary for some applications, such as when humans refer to the output to make decisions.

3.1.6 Quality in the whole system using machine learning

(Related to System Quality)

No matter how many man-hours are spent, the accuracy of the model can never be 100 percent, and achieving high accuracy is not the fundamental goal of system development. Also, in the case of unsupervised learning, it is not possible to use a relative performance indicator such as a correct answer and compare it with the correct answer. For this reason, it is necessary to define **Key Performance Indicator (KPI)**, which is a target indicator from the overall perspective of the business or system, and discuss quality from that perspective. For example, in the case of a web application, it is necessary to measure the user satisfaction. For example, if the KPI is to be user satisfaction or behavioral promotion, such as a web application, it is necessary to compare whether or not the system is deployed by using **A/B Testing**, and then to perform **Hypothesis Testing** to confirm that the KPI is improved by the deployment. **Hypothesis Testing** to evaluate the overall quality of the system.

3.2 Quality control in AI products

(Related to Process Agility)

Based on the characteristics of machine learning described in the beginning of this chapter, the following points should be kept in mind when managing the quality of the model and the entire system including the model.

- It is difficult to predict and agree on the quality (especially performance indicators such as accuracy) that can be ensured in advance, so a testing and exploratory process is adopted to find out to what extent the quality can be achieved through repeated experiments. In some cases, for example, it is necessary to adjust the requirements and use cases, and make a decision such as, "Even if the conformance rate is low, we will accept it if the reproduction rate is high. The understanding and cooperation of customers and other stakeholders is necessary for the testing and exploratory process.
- When the distribution of the data used for training and the data input during operation are different, the quality (especially performance indicators such as accuracy) may deteriorate. This occurs when assumptions made during development are inadequate, but it is difficult to make all assumptions when dealing with the real world such as automated driving. Furthermore, the behavior and preferences of customers, objects in the real world, and the characteristics of the camera that captures them all change, so even if the assumptions made at the time of development are sufficient, the performance may deteriorate at runtime. For this reason, it is necessary to set up a runtime monitoring mechanism to detect performance degradation. To make it easier

to identify problems, we should monitor not only the resulting performance, but also individual elements such as the range and distribution of input values and their relationships [Breck 2016].

- If we perform online learning, i.e., continuously update the model using runtime data as training data, we can adapt to the latest distribution. However, in this case, there is a possibility that inappropriate training data may result in an inappropriate model. When automating the continuous learning (model updating), it is necessary to automate the selection of training data, performance evaluation and testing as well.
- In a system using machine learning, it is the generated model that is part of the product, but the resulting model depends on the training data, the program that implements the learning algorithm and its hyperparameters. In order to rebuild the same model or to understand the circumstances under which it was built, it should be recorded to achieve reproducibility. For example, if options related to the learning algorithm are set on the command line, the information should be recorded so that it is not lost.

3.3 Quality assurance techniques for AI products

As for quality assurance techniques other than the performance point of view described in 3.1.1, testing techniques in particular are being actively researched [Zhang 2020], but they are still in the development stage and highly dependent on the application. It should be noted that new tools may be released in the future.

3.3.1 Pseudo-oracle

(Related to Model Robustness and System Quality)

For models obtained from machine learning and the systems that contain them, it is often not possible to clearly define the correct answer or expected value (test oracle) for all the various possible inputs, or defining them is expensive, requiring manual work. For this reason, the number of test cases tends to be limited. In order to set up test cases that try a large number and variety of inputs, we prepare **pseudo-oracle**to be compared. For example, compare with another implementation (N-version programming), an older version, or a rule-based implementation. Even if the output does not always match perfectly, defining an error/distance and examining the cases where the error/distance is large may help to notice errors and gain insights. It is also possible to use **Search-Based Testing**, which uses evolutionary computation to search for test cases where the error and distance are large [Pei 2017].

3.3.2 Metamorphic testing

(Related to Model Robustness and System Quality)

The accuracy of the model is not 100 percent, so even if there is an error in the output, it does not mean that there is an implementation failure such as a coding error. For this reason, testing cannot be done in the conventional way, where one is convinced of the existence of a defect if the output deviates from the expected value. In **metamorphic testing**, a test that can be judged as correct or incorrect is obtained by using the relationship that "if a certain change is given to the input, the change in the output can theoretically be expected" (metamorphic relationship) [Chen 2018]. If we are confident that the metamorphic relation is inherently valid, we can consider cases where the relation does not hold to be indicative of an implementation failure. Alternatively, the failure of an expected relation to hold may

indicate that our understanding of the implemented model was incorrect.

In metamorphic testing, the metamorphic relationships that are likely to be established are determined from patterns such as "adding or multiplying a constant value to an attribute of a data point," "adding or deleting a data point," and "replacing a data point".

Metamorphic testing is used in checking the robustness of models to input perturbations [Tian 2018] and can also be used to check training pipelines such as learning algorithms [Dwarakanath 2018].

3.3.3 Robustness testing

(Related to Model Robustness and System Quality)

In order to evaluate the robustness described in 3.1.3, we may add changes to the input test data and evaluate whether the output remains unchanged. For example, if the input is an image, we can consider changes in illumination, addition of rain or fog due to image composition, partial loss or distortion, etc. A number of tools have been studied [Tian 2018], such as search-based testing to search for cases where a change in the input would result in a larger change in the output.

3.3.4 Coverage in Neural Networks

(Related to Model Robustness)

In conventional software white-box testing, coverage metrics such as branch coverage have been used to evaluate how well the various situations contained in the implemented program have been tested. On the other hand, when neural network based learning algorithms (mainly deep learning) are used, the behavior of the implemented model depends on the size of numerical values rather than logical conditional branches. For this reason, even if branch coverage, etc., is used in the program representing the model, the coverage value will reach 100% with a little testing. In response to this, it has been considered that the numerical values of various computational components (neurons) in a neural network can be used to evaluate the diversity of tests and generate a variety of tests [Pei 2017] [Ma 2018]. However, it has been reported and discussed that there is not necessarily a strong correlation with the required quality [Harel-Canada 2020], and it should not be easily interpreted and applied, at least not blindly, as the coverage metric for conventional programs.

3.3.5 Technologies for explainability and interpretability

(Related to System Quality)

Explainability and interpretability techniques described in 3.1.4 [Gunning 2016] include, for example, those that provide explanations for individual outputs and those that provide explanations for the model as a whole. In the former case, information is presented about which parts of the input data had a significant influence on the decision for a given output. In the latter case, it uses a highly explainable IF-THEN rule form or a tree structure form (decision tree) to allow people to interpret what kind of regularities and criteria the entire model has acquired. We will show the trend of this technology in more detail in the next chapter.

3.4 References

〔Goodfellow 2015〕Ian J. Goodfellow, Jonathon Shlens, Christian Szegedy, Explaining and Harnessing Adversarial Examples, https://arxiv.org/abs/1412.6572 , v3, 2015 年

〔大場 2018〕 AI システムの品質保証の動向, SQuBOK Review 2018 Vol.3, pp. 1-12, 2018 年

〔石川 2019〕 石川 冬樹, 徳本 晋, 機械学習応用システムのテストと検証, 情報処理 Vol. 59 No.1, pp.25-33, 2019 年

〔Breck 2016〕 Eric Breck, Shanqing Cai, Eric Nielsen, Michael Salib, D. Sculley, What's your ML test score? A rubric for ML production systems, Reliable Machine Learning in the Wild - NIPS 2016 Workshop, 2016 年

〔EC 2019〕 European Commission, High-Level Expert Group on AI, Ethics Guidelines for Trustworthy Artificial Intelligence, https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai

〔Kuwajima 2019〕Hiroshi Kuwajima, Fuyuki Ishikawa, Adapting SQuaRE for Quality Assessment of Artificial Intelligence System, The 30th International Symposium on Software Reliability Engineering, pp.13-18, 2019 年

〔Mehrabi, 2019〕Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, Aram Galstyan, A Survey on Bias and Fairness in Machine Learning, https://arxiv.org/abs/1908.09635, 2019 年

〔Gunning 2016〕 David Gunning, Explainable Artificial Intelligence (XAI), https://www.darpa.mil/program/explainable-artificial-intelligence, 2016 年

〔Zhang 2020〕 Jie M. Zhang, Mark Harman, Lei Ma, Yang Liu, Machine Learning Testing: Survey, Landscapes and Horizons, IEEE Transactions on Software Engineering, 2020 年

〔Pei 2017〕 Kexin Pei, Yinzhi Cao, Junfeng Yang, Suman Jana, DeepXplore: Automated Whitebox Testing of Deep Learning Systems, The 26th Symposium on Operating Systems Principles, pp.1-18, 2017 年

〔Chen 2018〕 Tsong Yueh Chen, Fei-Ching Kuo, Huai Liu, Pak-Lok Poon, Dave Towey, T. H. Tse, Zhi Quan Zhou, Metamorphic Testing: A Review of Challenges and Opportunities, ACM Computing Surveys, Vol.51 No.1, 2018 年

〔Tian 2018〕 Yuchi Tian, Kexin Pei, Suman Jana, Baishakhi Ray, DeepTest: Automated Testing of Deep-Neural-Network-driven Autonomous Cars, The IEEE/ACM 40th International Conference on Software Engineering, pp.303-314, 2018 年

〔Dwarakanath 2018〕 Anurag Dwarakanath, Manish Ahuja, Samarth Sikand, Raghotham M. Rao, R. P. Jagadeesh Chandra Bose, Neville Dubash, Sanjay Podder, Identifying Implementation Bugs in Machine Learning based Image Classifiers using Metamorphic Testing, The 27th ACM SIGSOFT International Symposium on Software Testing and Analysis, pp.118-128, 2018 年

〔Ma 2018〕 Lei Ma, Felix Juefei-Xu, Fuyuan Zhang, Jiyuan Sun, Minhui Xue, Bo Li, Chunyang Chen, Ting Su, Li Li, Yang Liu, Jianjun Zhao, Yadong Wang, DeepGauge: multi-granularity testing criteria for deep learning systems, The 33rd ACM/IEEE International Conference on Automated Software Engineering, pp.120-131, 2018 年

〔Harel-Canada 2020〕Fabrice Harel-Canada, Lingxiao Wang, Muhammad Ali Gulzar, Quanquan Gu, Miryung Kim, Is neuron coverage a meaningful measure for testing deep neural networks?, The 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2020 年

4. Explainability and interpretability in machine learning

4.1 Introduction

This technical catalog is intended as a reference for AI product developers, testers, and quality assurance professionals who are required to be accountable for their work.

Machine learning is capable of prediction and recognition with high accuracy, but most of them do not explain the basis for their decisions. The concern that "if you can't explain what's in it, you can't use it" can lead to distrust of AI products, which can be a factor inhibiting their adoption. For this reason, there is a growing social need to explain the rationale behind AI products.

In Japan, the Ministry of Internal Affairs and Communications (MIC) has formulated the "Draft Principles for AI Utilization," which states that as a "principle of transparency," users of AI products should pay attention to the verifiability of the input and output of AI systems and the accountability of diagnostic results. In addition, the "principle of accountability" requires users of AI products to be accountable to their stakeholders. Overseas, the EU's General Data Protection Regulation (GDPR) imposes accountability in decision-making regarding users, and in the US, the XAI (Explainable AI) project at DARPA, the origin of the term, is underway.

The focus on research on explainability and interpretability of machine learning models has increased especially since 2016. The survey paper [4][3], which summarizes recent research, shows that research on explainability and interpretability will continue to increase. While there is a lot of research going on, it is difficult for users of explainability and interpretability to know which methods to use and when to use them. In this chapter, we organize and categorize the methods for providing explanatory and interpretive properties to machine learning models, and introduce the details of representative methods in the following sections.

Even if the machine learning model can explain the problem, the user must be convinced by the explanation. The explanation is only a means to convince the user. For this reason, we will focus on the areas where a machine learning model should not be a black box, and we do not intend to deny all black box models.

4.2 Classification of methods for adding explainability and interpretability

There are various ways to categorize the methods for adding explainability and interpretability to machine learning models. For example, according to Associate Professor Hara of the Research Institute of Industrial Science and Technology, Osaka University in the literature [2][1][6], there are three major exit-based classifications of "what kind of explanation do we want": "global explanation", "local explanation", and "explainable model design".

[Classification by Associate Professor Hara]

1. Global description
   A method of explaining a complex black-box model by representing it as a readable and explain-

able model.

2. Topical Description
   A method that serves as an explanation by presenting the rationale for the predictions of a black box model for a particular input. This includes explanatory methods for deep learning models, especially image recognition models.
3. Explainable Model Design
   A method of creating a highly readable and explainable model from the beginning.

In addition, Mitsubishi Electric Corporation classifies explainability and interpretability on three axes[5]。

[Classification by Mitsubishi Electric Corporation]

(1). The object to which explainability/interpretability is given.
(2). When to give explainability/interpretability.
(3). The method of providing explainability/interpretability.

The classification of (1) is based on the object to be explained or interpreted, and is defined as Global for methods that are applied to the model itself, and Local for methods that are applied to individual inference results. In contrast, the Local approach is to approximate the predictive trend globally to a simple model that can be interpreted. In contrast, Local is an explanation of individual prediction results.

The classification of (2) is based on when the explanatory and interpretive properties are given, and there are two types: Intrinsic and Post-hoc. Intrinsic refers to the approach of providing explanatory and interpretive properties to the internal properties and mechanisms of a model during model training, or training a model that can be explained and interpreted from the beginning, etc. Post-hoc refers to the approach of providing explanatory and interpretive properties to a trained model after model training. Explainability and interpretability methods can be broadly classified into four categories based on the combination of Global/Local and Intrinsic/Post-hoc. Furthermore, (3), Explainability and Interpretability (Explanator), can be classified into three major approaches (Rule-based, Feature-based, Instance-based). Rule-based is a method of explaining inference logic and rationale for decisions based on rules such as "if ..., then ...," while feature-based is a method of weighting important elements of input data that have a significant impact on inference. Instance-based is a method of explaining the inference results of a model using the instances contained in a data set and the data processed from those instances.

Figure 4.1 summarizes the representative methods using these classification axes and shows the correspondence with Associate Professor Hara's classification method. We will discuss each of the representative methods in detail later. It is very important to consider how to provide feedback to people, such as what kind of information should be presented to dispel their concerns and make them trust the AI's judgment.

Global's Intrinsic approach uses methods that have a clear inference process. Modeling methods such as the General Linear Model (GLM) and the Generalized Additive Model (GAM) provide transparency and interpretation of data based on the characteristics of the model. Decision trees (DTs) and decision rules (DRs) can be used to obtain interpretations according to the rules that have been set. Neighbor-based methods (Neighbor) explain the explanatory properties of individual layers after training them in a neural network by using the nearest neighbor method to predict the class of test samples.

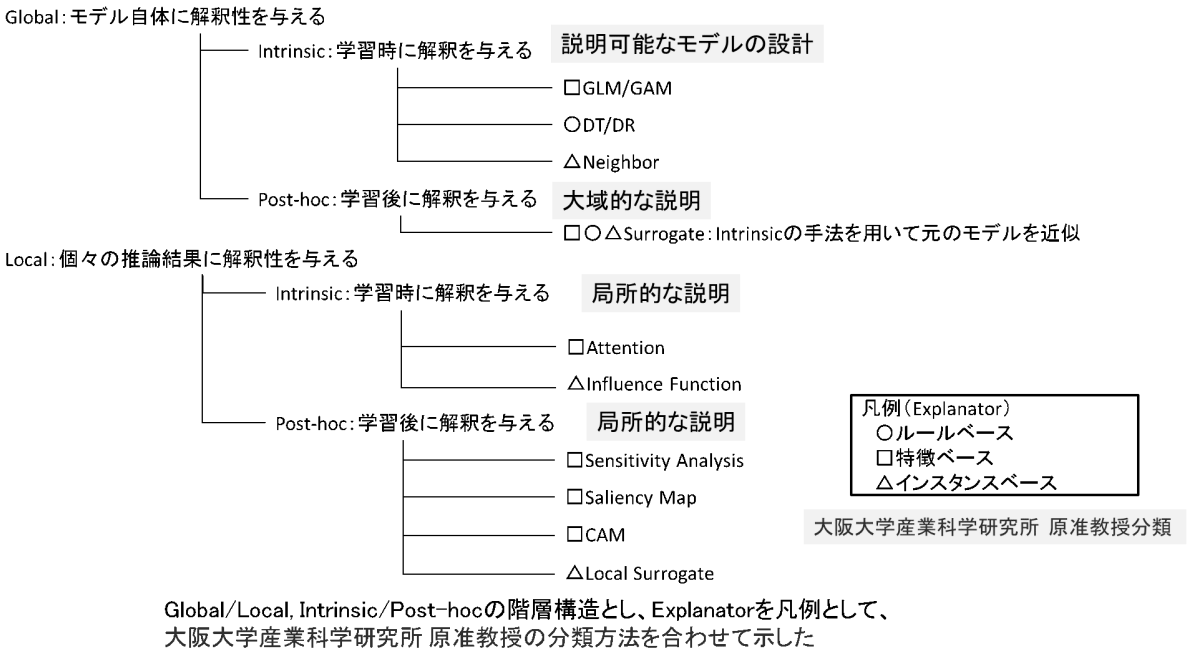


Fig. 4.1　Classification of explanatory and interpretive properties

Global's post-hoc approach is the Surrogate's method, which approximates the output of a model by interpretable methods. For example, DT Surrogate can explain the inner workings of a complex model by transforming it into a simple surrogate model using DT.

Local Intrinsic methods often use the properties of neural networks, and include approaches such as Attention, which uses an autoregressive model of feature maps to estimate points of interest. There is also an instance-based method that quantifies the influence of each training sample on the inference result using an Influence Function.

Post-hoc approaches include CAM (Class Activation Map), which estimates a saliency map based on gradients, and Sensitivity Analysis, which gives perturbations/variations to the data and analyzes the transformation of the output. There are also Local Surrogate approaches such as LIME, which uses artificial data created by perturbing the input data to learn interpretable models.

Table 4.1: Explanator

Explanator	Global /Local (G/L)	Instrinsic /Post-hoc (I/P)	Reference of typical methods

GLM/GAM	G	I	GLM/GAM Hastie, T. J.; Tibshirani, R. J. (1990). Generalized Additive Models. Chapman & Hall/CRC. ISBN 978-0-412-34390-2. Madsen, Henrik; Thyregod, Poul (2011). Introduction to General and Generalized Linear Models. Chapman & Hall/CRCC
DT/DR	G	I	DT/DR V. Schetinin et al., "Condent interpretation of Bayesian decision tree ensembles for clinical applications," IEEE Trans. Inf. Technol. Biomed., vol. 11, no. 3, pp. 312319, May 2007.
Neighbor	G	I	KNN(k-nearest neighbor) N. Papernot, P. McDaniel, Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning (2018). arXiv:1803.04765.
Surrogate	G	I	Surrogate Model J. J. Thiagarajan, B. Kailkhura, P. Sattigeri, and K. N. Ramamurthy. (2016). "TreeView: Peeking into deep neural networks via feature-space partitioning." [Online]. Available: https://arxiv.org/abs/1611.07429
Attention	L	I	Attention Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems. 2017.
Influence Function	L	I	Influence Function Pang et al., Understanding Black-box Predictions via Influence Functions, arXiv:1703.04730, 2017.
Sensitivity Analysis	L	P	Gradient Boost Machine Friedman, Greedy Function Approximation: A Gradient Boosting Machine, Annals of statistics, vol. 29, no. 5, pp. 1189-1232, 2001.
Saliency Map	L	P	Saliency Map Karen et al., Deep Inside Convolutional Networks: VisualisingImage Classification Models and Saliency Maps, arXiv:1312.6034v2, 2014.
CAM	L	P	Grad-CAM R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. arXiv preprint arXiv:1610.02391, 2016.

Local Surrogate	L	P	LIME M. T. RiPython や R で XAI の代表的な手法が使えるようにツールの整理も進んでいる beiro, S. Singh, and C. Guestrin, "'Why should i trust you?': Explaining the predictions of any classier," in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2016, pp. 11351144.

The XAI techniques mentioned above. In addition to the implementations published individually by the authors of each paper, a library that collects and organizes the various methods is also available.（Table4.2、Table4.3）.

Table 4.2: XAI related libraries

Name	Language	Description	Links
ELI5	Python	Python libraries, various description and visualization methods are implemented. It is designed to be seamlessly connected to scikit-learn, which is often used for machine learning in python.	https://eli5.readthedocs.io/en/latest/
iml	R	The R package, the book Interpretable Machine Learning: A Guide for Making Black Box Models Explainable.	https://github.com/christophM/iml
DALEX	R	The R package, a separate repository for package developers, has compiled a variety of information, including papers.	https://github.com/pbiecek/DALEX

Table 4.3: XAI-related Github

No.	Description	Links
1	The Institute for Ethical AI & A list of AI-related technologies compiled by Machine Learning. Explaining Black Box Models and Datasets contains information on libraries and Git.	https://github.com/EthicalML/awesome-production-machine-learning#explaining-black-box-models-and-datasets

2	The H2O.ai Machine Learning Interpretability team (https://github.com/h2oai/mli-resources) has a list of AI-related technologies based on machine learning workflows. Explainability- or Fairness-Enhancing Software Packages contains information on libraries and Git.	https://github.com/jphall663/awesome-machine-learning-interpretability/blob/master/README.md

4.3 Typical methods for adding explainability and interpretability

4.3.1 GLM/GAM

Table 4.4: GLM/GAM

Overview	A generalized linear model (GLM) is a type of machine learning model in which the objective variable is represented as a linear combination of explanatory variables. GAM (Generalized Additive Model) is a more generalized version of GLM. Both models are characterized by the ease of understanding the impact of each explanatory variable on the target variable.
Classification	Global, Intrinsic
Target domain	theoretically applicable to any kind of data. Theoretically, it can be applied to any data. However, it is not often used for images because it is difficult to visualize and the accuracy is low.
Target model	Both GAM/GLM are interpretable models in their own right.
Example of practical use	GAM has been implemented and is available in the data science platform DataRobot.
How to use	A GLM is a model in which explanatory variables are weighted by coefficient parameters and added together linearly. By analyzing the parameters of the model that has been trained on the training data, the sensitivity of each explanatory variable to the target variable can be determined. For example, if the coefficient w1 of an explanatory variable x1 is positive, we know that the value of the objective variable y increases as x1 increases. Alternatively, if w1 is zero, we can say that x1 has no effect on the value of y. The GAM is a model in which the explanatory variables are transformed by some function and added together in a linear fashion. As in the GLM, the sensitivity of each explanatory variable to the target variable can be determined by analyzing the parameters of the model.

Effect：Predictive Accuracy	GLM or GAM allows us to know the relationship between explanatory and objective variables, but if a relationship appears that is different from the relationship assumed by humans, it may be possible to detect and relearn overlearning to outliers, etc.
Effect: Reliability	If the results of the parameter analysis of the model show the same relationship between the explanatory variables and the objective variable that humans assume, it is considered to be highly reliable.
Concerns	GLM is a very simple model, which is easy to interpret but may not be accurate in some cases. GAM transforms the explanatory variables by some function, and the complexity of this function determines the trade-off between explanatory power and accuracy of the model. The complexity of the function determines the trade-off between explanatory power and accuracy of the model, and the complexity of the function needs to be adjusted by parameter tuning to obtain the desired explanatory power and accuracy.
library	**pyGAM (GAM library for Python)** https://pygam.readthedocs.io/en/latest/ **mgcv (GAM library for R)** https://cran.r-project.org/web/packages/mgcv/index.html
References	[4.3.1-1] J Nelder, R Wedderburn, “Generalized Linear Models”. Journal of the Royal Statistical Society. Series A 135（3）: 370 – 384. [4.3.1-2] T Hastie, “Generalized Additive Models”, Statistical models in S, 2017, Chapter 7.

4.3.2 DT

Table 4.5: DT

Overview	DT (Decision Tree) is an analysis technique that divides data in stages and outputs the analysis results in a tree structure. It is a method that outputs models that can be interpreted from the beginning, and is highly convenient for users in terms of providing interpretability to machine learning models.
Classification	Global, Intrinsic
Target domain	It is used for prediction, discrimination, and classification of data.
Target model	DT is a model that can be interpreted in its own right.
Example of practical use	It is implemented and available in DataRobot and various other tools.

How to use	1) According to the purpose of data analysis, the data is divided by the algorithm of decision tree analysis. 2) In a data partitioning situation, the quality of data partitioning is judged by the impurity (which indicates whether the data has been partitioned cleanly; the pure state is 0 (zero)) and the information gain (which indicates the goodness of the partitioning; it is obtained by the impurity before partitioning - the impurity after partitioning). 3) Limit the depth of the tree structure so that generalization performance can be ensured.
Effect: Prediction accuracy	The results of the analysis can be easily interpreted because they are expressed in a tree structure. Therefore, it is easy to judge the validity of the obtained machine learning prediction model.
Effect: Reliability	It is highly reliable because of its high readability due to its tree structure representation.
Issues of concern	If the tree structure becomes too deep, there is a risk of over fitting. Therefore, it is necessary to limit the depth of the tree structure to ensure the generalization performance.
Library	**dtreeviz（Visualization Tools）** https://github.com/parrt/dtreeviz **scikit-learn （Machine learning library for Python）** https://github.com/scikit-learn/scikit-learn **rpart, partykit （Library for R）** https://cran.r-project.org/web/packages/rpart/index.html https://cran.r-project.org/web/packages/partykit/index.html
References	[4.3.2-1] V. Schetinin et al., "Condent interpretation of Bayesian decision tree ensembles for clinical applications," IEEE Trans. Inf. Technol. Biomed., vol. 11, no. 3, pp. 312319, May 2007.

4.3.3 Surrogate

Table 4.6: Surrogate

Overview	Surrogate re-trains the Black Box model as an interpretable model from scratch. Global's Surrogate is a representation of what values of what variables the model is focusing on as a whole. Global Surrogate represents what values of which variables the model focuses on as a whole.
Classification	Global, Post-hoc
Target domain	Image, text.
Target model	Machine learning, deep learning.
Example of practical use	None.

How to use	In decision tree proxy models in decision trees are created by training a decision tree with the original inputs and predictions of the original complex model. Individual Conditional Expectation (ICE) and Partial Dependence (PD) are used to find and check interactions between variables.
Effect: Prediction accuracy	The decision tree proxy model allows us to read the importance of variables, trends and interactions. In the decision tree proxy model, the model itself has a hierarchical structure represented by equations and inequalities.
Effect: Reliability	When it is consistent with expectations, it leads to improved credibility for the user.
Issues of concern	A simple model does not fully represent the internal mechanisms of a complex model.
library	under investigation
References	[4.3.3-1] J. J. Thiagarajan, B. Kailkhura, P. Sattigeri, and K. N. Ramamurthy. (2016). “TreeView: Peeking into deep neural networks via feature-space partitioning.” [Online]. Available: https://arxiv.org/abs/1611.07429

4.3.4 TCAV

Table 4.7: TCAV

Overview	CAV (Concept Activation Vectors) is a method for quantifying how much a defined concept influences the inference of a classifier, and TCAV (Testing with Concept Activation Vectors) is a testing technique applied to a data set.
Classification	Global (TCAV: Whole Model), Local (CAV: One Variable), Post-hoc
Target domain	The paper covers image processing. It can be used to quantify how strongly the input image data follows a pre-defined concept.
Target model	Machine learning models (DNN, SVM, etc.) to which the target explanatory method can be applied.
Example of practical use	None.

How to use	It is assumed that the usual machine learning model development has been completed and a trained classifier is available. 1) Prepare a user-defined "concept" example and a random example that differs from the "concept" 2) Input the concept example and the random example into the DNN and use the compressed features propagated forward to an arbitrary intermediate layer as the respective feature values 3) Find a hyperplane that separates the feature space into two labels, concept and random, and use this as the CAV 4) Input the data that will be the target of the analysis and acquire the data in the same intermediate layer as 2. This is the sensitivity, and the inner product of this and the CAV is the Conceptual Sensitivity. 5) For the data set to be evaluated, calculate the degree to which the signal transmitted to the target class becomes stronger when the data is moved in the CAV direction. This is defined as the TCAV value. 6) The higher the TCAV value, the more the class can be judged to be in line with the set concept.
Effect: Prediction accuracy	Based on the results of TCAV, training data and models can be modified to improve prediction accuracy. For example, if the TCAV results show that a concept that should not have influenced the inference is influencing the inference, adding new data to the training data may improve the prediction accuracy.
Effect: Reliability	It is possible to quantify whether the developed model is acquiring the expected generalized knowledge (close to human perception). This will be used to enable validation of the model. It is also possible to show that there are no negative effects of bias by validating concepts that dare to be biased. (In the paper, we have also tested this by including text in the image.)

Issues of concern	Although this is a common concern with CAM and other systems, there is no way to uniquely determine which layer to select for feature extraction, and it is experimental to determine which layer of extracted information is truly the correct interpretation. In addition, the following two points are of limited applicability. **Formation of conceptual data** The data we deal with needs to be able to form concepts that are uniquely interpreted by humans. The paper focuses on images, and shows, for example, how a zebra could be decomposed into a horse and the concept of stripes. When we think about Natural Language Processing (NLP), which has developed rapidly in recent years, concept formation is not as simple as for images. **Model constraints** As described in the "How to Use" section, the model should be such that it extracts the final classification signal while applying some compression to the input. When applied to a simple model such as a decision tree, it is necessary to consider how to set its features to be appropriate.
Library	https://github.com/tensorflow/tcav
References	[4.3.4-1] Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors （TCAV）, Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Jun, Cai James, Wexler Fernanda, Viegas Rory, Abbott Sayres, ICML 2018.

4.3.5 Attention

Table 4.8: Attention

Overview	Attention mechanism is a method for learning the relationship between input and output elements and the points to pay attention to, which is introduced in deep learning models mainly for machine translation and image processing.
Classification	Local, Intrinsic
Target domain	Machine translation, image processing.
Target model	DNN
Example of practical use	The Attention mechanism is currently at the research level and is expected to be applied to areas such as image processing, automated driving [4.3.5-4] and machine translation [4.3.5-3].

How to use	In machine translation using the Attention mechanism, a new context vector is calculated using the hidden state of each word in the encoder section, which encodes the input string, and the hidden state of the target word translation in the decoder section, which is used for decoding [4.3.5-3]. The context vector is used for word estimation in the decoder section, and the Attention mechanism can be used to improve translation accuracy in long sentences and also produce useful results in the field of machine translation, called alignment, which is the process of analyzing the contrast between the pre- and post-translation sentences.
Effect: Prediction accuracy	Machine translation can show the relationship between words and text, and can analyze the grammatical structure of a language. For automatic driving and image processing applications, the relevance of pixels in the forward image to driving operations can be visualized, and their importance to the driving scene can be intuitively evaluated.
Effect: Reliability	Attention can visualize the relationship between each input and output element, which leads to improved reliability for users.
Issues of concern	By using the Attention mechanism, it is now possible to quantitatively analyze the correspondence between input and output data. One example is the ability to analyze the correspondence between words translated by machine translation and input words. However, when developing quality assurance technology, the correspondence between input and output data may not be as clear as in machine translation. For example, in the task of detecting failures of sensors mounted on factory machines, it is possible to predict sensor values using the Attention mechanism and analyze the relevance of input/output sensor data, but there is a debate as to whether the relevance can be explained to humans. However, it is debatable whether the relevance can be explained to humans.
Library	https://www.tensorflow.org/tutorials/text/nmt_with_attention

References	[4.3.5-1] Mikolov, Tomáš, et al. "Recurrent neural network based language model." Eleventh annual conference of the international speech communication association. 2010. [4.3.5-2] Gehring, Jonas, et al. "Convolutional sequence to sequence learning." Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 2017. [4.3.5-3] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems. 2017. [4.3.5-4] Liu, Nian, Junwei Han, and Ming-Hsuan Yang. "Picanet: Learning pixel-wise contextual attention for saliency detection."Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018. [4.3.5-5] Wang, Dequan, et al. "Deep object-centric policies for autonomous driving." 2019 International Conference on Robotics and Automation （ICRA）. IEEE, 2019. [4.3.5-6] https://qiita.com/itok_msi/items/ad95425b6773985ef959 [4.3.5-7] http://www.thothchildren.com/chapter /5c0b968d41f88f26724a70b8

4.3.6 Sensitivity Analysis

Table 4.9: Sensitivity Analysis

Overview	Sensitivity Analysis is an analytical method to understand the degree to which the presence of uncertain factors among important factors affects the overall outcome by varying these values. It has been used in various fields for many years, including economics, mathematics, medicine, and systems analysis. When sensitivity analysis is applied to AI, data can be intentionally varied and simulated to evaluate if the accuracy, fairness, security, and stability of the AI model's behavior and output predictions are acceptable. The evaluation results can also be used to provide a local description of the AI model.
Classification	Local, Post-hoc
Target domain	It can be used as one of the verification methods to check whether the implementation of AI models meets the design intent. In particular, it is effective in verifying the possibility that a small change in input data may result in a large change in the predicted output. It can also be used as a debugging method for AI models.
Target model	Since this method is model-independent, it can be widely used.
Example of practical use	It is a method that has already been put to practical use in a variety of fields.

How to use	1) Select key data that have explanatory or fairness implications. For example, LIME can be used as a reference for selecting important data. 2) Determine acceptable thresholds for acceptable changes in explanatory or fairness values. It is necessary to confirm in advance that the acceptable threshold is reasonable. 3) Initiate monitoring for unacceptable changes in explanatory or fairness values by manually or automatically varying the input data. 3a) If the changes in these values are within acceptable thresholds, then the explanation and fairness techniques are stable and not a problem. 3b) If the change in these values exceeds the acceptable threshold, debug and determine the cause, and then improve the AI model.
Effect: Prediction accuracy	**Improving explainability** Showing how the behavior and output of the model changes over time can enhance understanding and explanatory power. It can also be used for local explanation by showing the behavior and output of the model for a particular change in data. **Remarks** Unlike traditional linear models, machine learning algorithms generate very complex nonlinear, non-monotonic response functions, and the correlations between input and target variables are complex. Therefore, rather than searching static training data for hidden correlations, it is easier to focus on the potential instability of model prediction by utilizing sensitivity analysis methods for a quick and appropriate check.
Effect: Reliability	If it has been fully validated by sensitivity analysis, the AI model of interest is in a position to adhere to the domain knowledge and expectations of humans and is sufficiently reliable. In this case, the behavior and output of the model can be stabilized even if the data is slightly corrupted as expected.

Issues of concern	In the case of machine learning, it is better not to focus too much on numerical instability of model parameters. Since machine learning algorithms produce very complex nonlinear, non-monotonic response functions, traditional linear model validation methods are not suitable. It is difficult to determine which input data should be varied, how and by how much. This is because it is wrapped in a black box, and the relationship between the variation factors (parameters) and the results is unclear. A small change in the value of an input variable can result in a large change in the predicted response. For example, it is desirable to determine the range of variation for each individual variable after discussing the real possibility among the parties concerned, rather than mechanically taking a range of 20
Library	**Cleverhans** https://oreil.ly/3038mur https://github.com/tensorflow/cleverhans **Foolbox** https://oreil.ly/31NsDoD **What-If Tool** https://oreil.ly/2KJGHZ7 https://pair-code.github.io/what-if-tool/ **Interpretable Machine Learning with Python** https://oreil.ly/33xjthx
References	[4.3.6-1] An Introduction to Machine Learning Interpretability , 2nd Edition - An Applied Perspective on Fairness, Accountability, Transparency, and Explainable AI - https://www.oreilly.com/library/view/an-introduction-to/9781098115487/ [4.3.6-2] 観察研究における感度分析の勧め 入門編 ・・・観察研究に基づく意思決定に関わる全ての方へ・・・ http://www.jpma.or.jp/medicine/shinyaku/tiken/allotment/sensitivity_analysis.html http://www.jpma.or.jp/medicine/shinyaku/tiken/allotment/pdf/sensitivity_analysis.pdf

4.3.7 CAM

Table 4.10: CAM

Overview	CAM (Class Activation Mapping) is a method to visualize the basis for decisions in image analysis and classification using CNN (Convolution Neural Network).
Classification	Local, Post-hoc

Target domain	Image。
Target model	DNN。
Example of practical use	Grad-CAM (Gradient-weighted Class Activation Mapping) is used by Hitachi, Ltd. as a tool for engineers and consultants to explain the rationale for AI decisions to client company field personnel and to improve AI models as needed. . https://xtech.nikkei.com/atcl/nxt/news/18/06939/
How to use	For each feature map after feature extraction calculation in the CNN layer, GAP (Global Average Pooling), or pixel average value, is calculated and mapped to the class of classification. As a result of the mapping, weights for each feature map are output, so each feature map is weighted and overlaid on the original image to create a heat map showing the contribution of the mapping results.[4.3.7-1]
Effect: Prediction accuracy	Since the system can show where in the image it paid attention to perform the classification, it is possible to verify the quality. Even if the classification is successful, if the region of interest is not appropriate, the system will be able to make decisions such as modifying the learning method, which will lead to improved prediction accuracy.
Effect: Reliability	CAM can visualize the rationale for decisions, which leads to increased credibility for users.
Issues of concern	Because CAM uses GAP and Softmax functions, it is less accurate than the base image classification model. In addition, various methods such as VQA (Visual Question Answering)[4.3.7-2], which adds captions to feature regions and asks and answers questions about images, have been developed. Answering), which asks and answers questions about the image. It is also known that many of these methods are vulnerable to hostile attacks, and that by placing invisible noise in the image, it is possible to change the point of interest as a basis for judgment.[4.3.7-3]
Library	https://github.com/jazzsaxmafia/Weakly_detector
References	[4.3.7-1] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, Antonio Torralba, Learning Deep Features for Discriminative Localization, Computer Vision and Pattern Recognition, 2015. [4.3.7-2] Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization, Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra, Computer Vision and Pattern Recognition, 2016. [4.3.7-3] Ian J. Goodfellow, Jonathon Shlens, Christian Szegedy, Explaining and Harnessing Adversarial Examples, Computer Vision and Pattern Recognition, 2015.

4.3.8 LIME

Table 4.11: LIME

Overview	LIME (locally interpretable model-agnostic explanations) is a method to visualize the basis for decisions by approximating complex models with linear regression.
Classification	Local, Post-hoc
Target domain	Image, text.
Target model	Linear regression.
Example of practical use	Research phase.
How to use	A linear regression model is created using the data set obtained by repeated sampling and prediction from the data space around the target sample as the teacher data.
Effect: Prediction accuracy	The system will be able to show where the system focused its attention on the images and text for classification, which will allow for quality verification. Even if the classification is successful, if the region of interest is not appropriate, the system will be able to make decisions such as modifying the learning method, which will lead to improved prediction accuracy.
Effect: Reliability	LIME can visualize the basis for decisions, which leads to increased credibility for users.
Issues of concern	Since LIME is a local approximation, it may deviate from the expected result if the input data is far apart in the feature space.
Library	https://github.com/marcotcr/lime
References	[4.3.8-1] Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin, "Why Should I Trust You?": Explaining the Predictions of Any Classifier, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August, 2016.

References

[1] 【記事更新】私のブックマーク「機械学習における解釈性」. https://www.ai-gakkai.or.jp/my-bookmark_vol33-no3/.
[2] 【記事更新】私のブックマーク「説明可能 *AI*」(*Explainable AI*). https://www.ai-gakkai.or.jp/my-bookmark_vol34-no4/.
[3] A. ADADI et al: “Peeking inside the black-box: A survey on Explainable Artificial Intelligence（XAI）.” In: VOLUME XX, 2018.
[4] Guidotti et al. “A Survey of Methods For Explaining Black Box Models.” In: arxiv, 2018.

[5] 瀬光孝之 他. “機械学習モデルの解釈性に関する最新動向”. In: **電子情報通信学会**. Vol. Vol.102, No.10. 2019/10. URL: https://app.journal.ieice.org/trial/102_10/k102_10_973/index.html.

[6] **機械学習モデルの判断根拠の説明**. https://www.slideshare.net/SatoshiHara3/ss-126157179?qid=91472032-d83b-4d83-9305-a60e80f3aed9&v=&b=&from_search=4.

5. Generative systems

5.1 Assuming system

The target is a system that generates contents such as texts, conversations, images, videos, etc. that realize some kind of "goodness" perceived by humans, such as creativity, naturalness, and interestingness. In such systems, a technique called generative modeling is used, which learns "what kinds of things naturally exist in what kind of tendency (distribution)" (in contrast to discriminative models used in image recognition, which learn only "differences and boundaries of things"). In the same way that image recognition has been applied to the industry through technological evolution based on deep learning, content generation is likely to be applied to the industry in the future. On the other hand, in these systems, it is important to achieve human sensory satisfaction, and it is difficult to evaluate the quality objectively or automatically. In this chapter, we discuss quality assurance approaches for content generation systems that use generative models and are specifically evaluated by human senses.

The following types of such systems have already been commercialized or researched and developed.

- image and video generation systems: Generating natural images, illustrations, and targeted animations, and generating new images and videos based on rough posture specifications and attributes.
- Text-to-speech and voice conversion system: Generating entertaining voices with the atmosphere and personality of a specific character.

5.1.1 Application areas covered in this chapter

In this chapter, we consider systems that generate images and videos as representative examples of the systems described above. For example, the system can be used to generate and place images that make viewers feel happy on websites and in printed materials, or to generate videos that are the building blocks of movies, games, and animations. Examples of such systems are presented below.

example of image generation - with nothing specified

Generation of diverse and natural images without any specification (unsupervised learning).

Reference example (Figure 5.1): A Style-Based Generator Architecture for Generative Adversarial Networks. Tero Karras, Samuli Laine, and Timo Aila. In. CVPR 2019.

5.1.2 Example of image generation - image generation with class specification

Generate diverse and natural images by specifying the class of the image.

Reference case (Figure 5.2): Large Scale GAN Training for High Fidelity Natural Image Synthesis. Andrew Brock, Jeff Donahue, and Karen Simonyan . In ICLR 2019.

Fig. 5.1　Example of image generation - without specification (Karras et al., in CVPR 2019)

Figure 6: Additional samples generated by our model at 512×512 resolution.

Fig. 5.2　Example of image generation - image generation with type (Brock et al., in ICLR 2019)

5.1.3　Example of image generation - background generation by specifying reference image and layout

Given an art image and layout specification (object area placement), generate a background with the specified layout, reflecting the detailed texture of each area of the reference image.

Reference example (Figure 5.3): Anime Generation with AI. DeNA, 2020. `https://www.slideshare.net/hamadakoichi/anime-generation-ai`

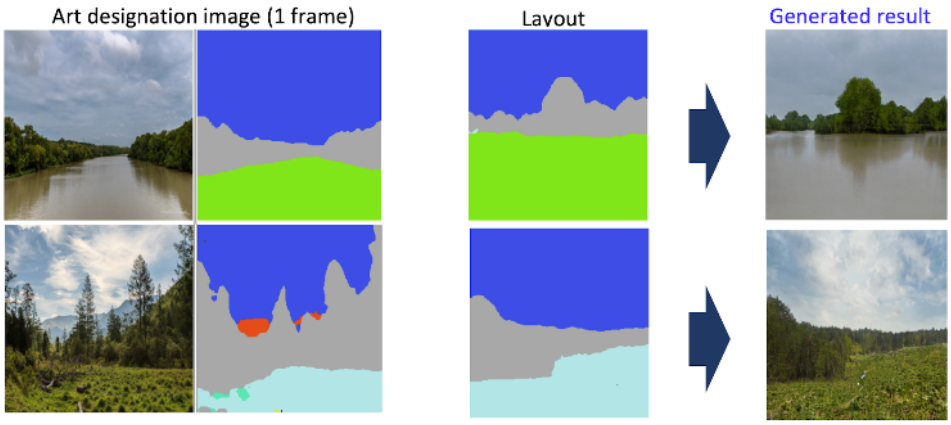


Fig. 5.3　Example image generation - background generation with reference image and layout (DeNA, 2020)

5.1.4　Example of image generation - line drawing coloring with color swatches

Specifying color swatches, line drawings, and rough part areas, and coloring them to strictly reflect the color patterns and line details of each part.

Reference example (Figure 5.4): Anime Generation with AI. DeNA, 2020. `https://www.slideshare.net/hamadakoichi/anime-generation-ai`, `https://youtu.be/X9j1fwexK2c?t=191`

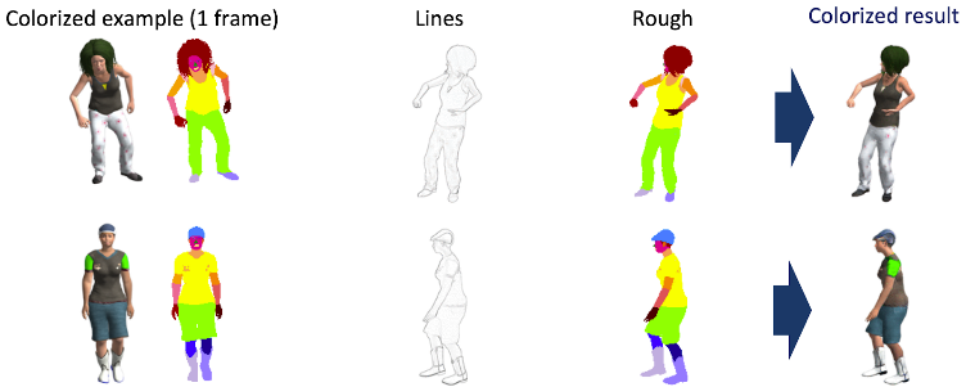


Fig. 5.4　Example of image generation - line drawing coloring with color swatches (DeNA, 2020)

5.1.5　Video generation example - specifying character image and structure sequence

Generate a video by providing a sequence of structural information that represents the pose you want the character to take.

A sequence of coordinate models (a list of postures) representing the poses to be taken by the character is given, and a natural animation of various characters moving in the specified poses is generated.

Reference example (Figure 5.5): Full-body High-resolution Anime Generation with Progressive Structure-conditional Generative Adversarial Networks . Koichi Hamada, Kentaro Tachibana, Tianqi Li, Hiroto Honda, and Yusuke Uchida. In ECCV Workshop 2018. https://youtu.be/bIi5gSITK0E https://youtu.be/0LQlfkvQ3Ok

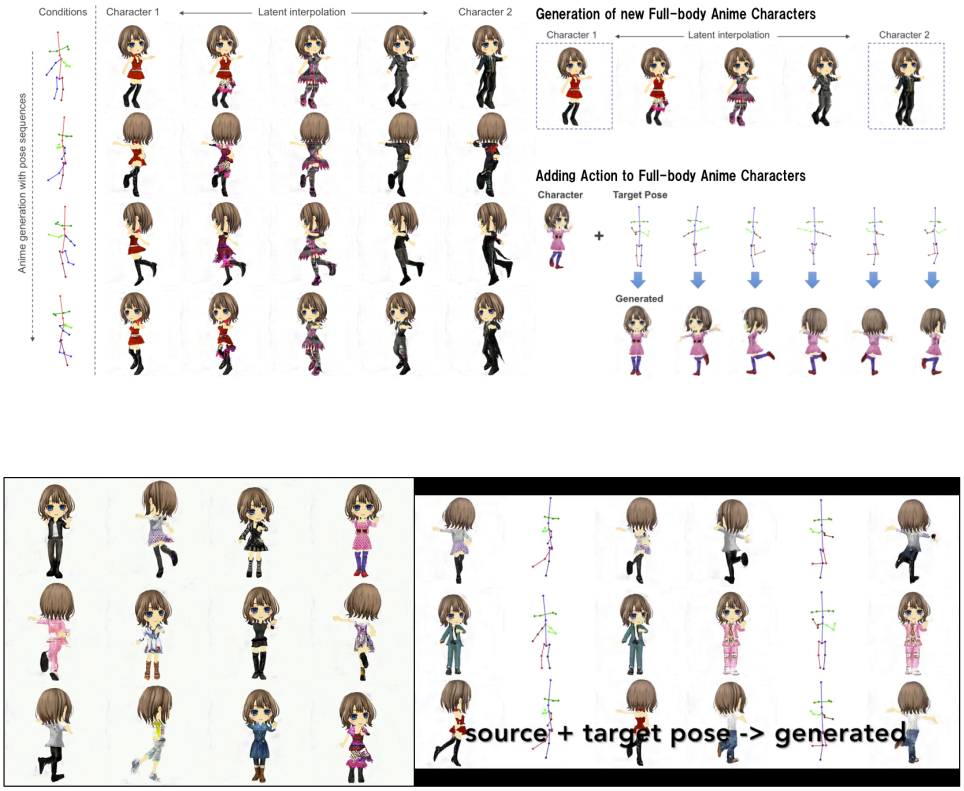


Fig. 5.5 Video generation example - specifying character image and structural sequence (Hamada at el., in ECCV Workshop 2018)

Given a sequence of shapes for each part of the body (a list of region information) for which we want to draw a character, we generate a natural animation in which various characters move in the order of the shapes and postures of each specified part.

Reference case (Figure 5.6): Anime Generation with AI. DeNA, 2020. https://www.slideshare.net/hamadakoichi/anime-generation-ai https://youtu.be/X9j1fwexK2c?t=166

5.1.6 Video generation example - middle section generation with keyframes

Generate natural and diverse videos by giving the starting and ending images as keyframes and "interpolating" between them.

Reference case (Figure 5.7): The challenge of animation generation by AI. Koichi Hamada, Tianqi Li. In DeNA TechCon 2019. https://www.slideshare.net/hamadakoichi/anime-generation https://www.slideshare.net/hamadakoichi/anime-generation-ai https://youtu.be/tOZW_KWb8b0 https://youtu.be/X9j1fwexK2c

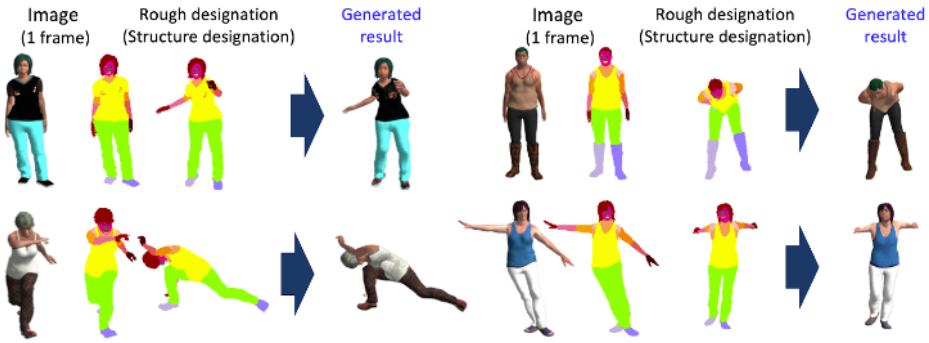


Fig. 5.6 Video generation example - specifying character image and structure sequence (DeNA, 2020)

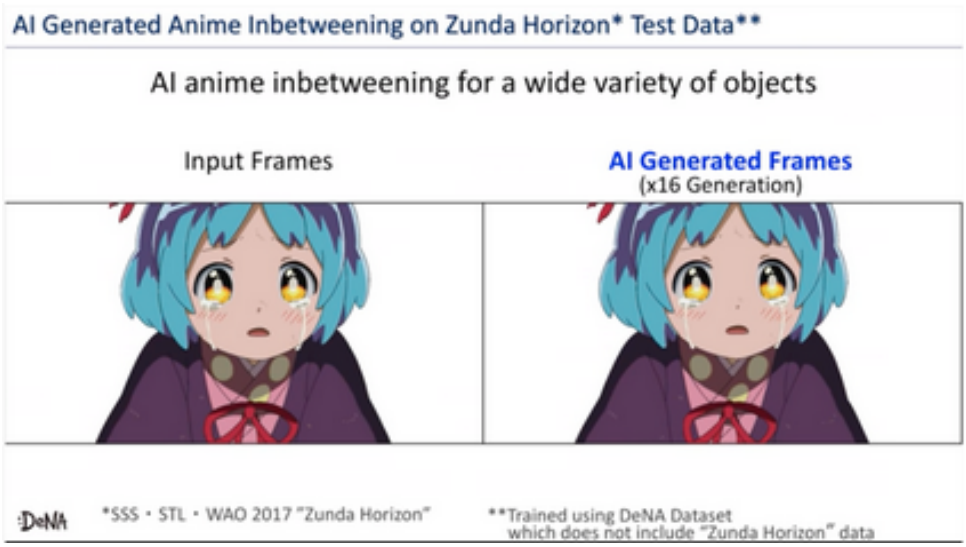


Fig. 5.7 Video generation example - specifying character images and structural sequences (DeNA, 2019)

5.1.7 Background - recent progress in AI-based generative models -

In recent years, the development of various methodologies for deep generative models has greatly opened up the possibility of AI-powered generation. In particular, advances in Generative Adversarial Networks (GANs) [Goodfellow+14] since 2018 [Karras+18,Brock+19,Karras+19] have led to high-quality generation that can be mistaken for the real thing. In the past, it was difficult to generate complex structures such as the whole body of a person, and high-quality generation was limited to rigid bodies such as vehicles and rooms, or only some parts of the body such as the face [Karras+18,Brock+19,Karras+19]. However, recent advances in technology that enables both high quality generation and structural integrity [Hamada+18] have made it possible to generate complex structures such as entire characters with high quality, and this has opened up the possibility of applications in the real industry of animation and games, such as the generation of illustrations and animations of various characters [Hamada+18, Hamada+18, Hamada+19].

GANs [Goodfellow+14] is a generative model that improves the quality of generation by pitting a

Generator against a Discriminator. The discriminator discriminates between real data and the data generated by the generator, and the generator tries to make the discriminator misidentify the generated data as real data in order to improve the quality of generation. When the discriminator and the generator mutually improve and reach an equilibrium point (Nash equilibrium), it is guaranteed that the distribution of the generated data by the generator matches that of the real data. However, learning GANs is difficult, and it is particularly important to have a methodology that stabilizes the learning of high-resolution, high-quality generation for a wide variety of targets.

Karras et al. [Karras+18] solved this problem by starting with a 4x4 low-resolution image and letting the generator and discriminator fight it out, then progressively increasing the resolution and growing the generator and discriminator to produce high-quality, diverse face images at 1024x1024 resolution.

In terms of generation by specifying the type of target, the quality of generation by specifying 1000 classes of ImageNet [Russakovsky+15] has been greatly improved [Odena+17,Miyato+18,Zhang+18,Brock+19]. Brock et al. [Brock+19] have achieved high-quality generation of 1000 classes at 512x512 resolution by learning in large batch sizes and channels using various learning stabilization methodologies.

Progress in generating complex structures such as the whole human body is also made [Hamada+18, Hamada+19, Hamada+20]. That is important for industrial applications but had remained a challenge [Karras+18, Brock+19, Karras+19]. Hamada et al. [Hamada+18] have achieved both high quality generation and structural consistency by progressively learning structurally conditioned GANs from 4x4 resolution, and have achieved high quality generation for complex structures such as the whole body of a character at 1024x1024 resolution. A variety of character animations can be generated, including image generation by specifying the coordinates of the pose to be taken by the character, and video generation by specifying the generation sequence. In addition, in the generation of natural and diverse animations, such as "inbetweening" (generating intermediate frames) between frames given a starting frame and an ending frame, which is performed in animation production, we have achieved to generate inbetweens between frames with large structural changes, which has been difficult in the past, by using structurally conditioned generation learning [Hamada+19, Hamada+19, Hamada+20]. The above progress has opened up a wide range of possibilities for the practical application of generative models, such as the generation of illustrations and animations of various characters for real industrial applications in animation and games [Hamada+18, Hamada+19, Hamada+20].

5.1.8 Expected Use Cases and Input/Output

In the example of this guideline, we consider an example that assumes industrial applications of animation and games, such as the generation of illustrations and animations.

In the discussion of this example, the image/animation to be generated must be an illustration, not a live action, and must be a human character. Also, we do not consider that the system automatically acquires training data, updates the model, and deploys it. In other words, we do not consider that the system will continue to learn from an unspecified large amount of uncontrolled data, such as on the Web. We assume that the addition of training data and the updating of models will be done under the direction of the developer, and that quality evaluation will be done by the developer.

In the development of the generative model, the primary goal is to generate content that is close to what exists in nature. The function that can be directly realized from this is to generate a large number and variety of natural contents. On the other hand, in industrial applications, we may want to add constraints to the content to be generated. For example, we may want to generate a video in which a specific animated character moves in a certain way. For this reason, in the examples in this chapter, we will consider different cases regarding whether or not to specify "what kind of image/video to generate"

and how to specify it.

In the following, we describe the five use cases. UC-0 is the most basic and direct use of the generative model, whereas UC-1A/2A require the user to follow instructions on the structure (posture) of the character in the image/video, respectively, and UC-1B/2B require the user to follow instructions on the attributes of the character in the video/image, respectively (to be precise, UC-2B includes rough instructions also for posture). For these use cases, it is necessary to produce output that follows the specifications for each structure and character, and it is necessary to handle the characteristics according to these specifications in addition to the quality characteristics handled by UC-0. In addition, the video generation of UC-2A/2B needs to be extended to take into account the quality characteristics handled in the image generation of UC-1A/1B.

[Use Case UC-0: Unspecified, Image]

Generate diverse and natural images without specifying anything.

[Use Case UC-1A: structure specification, image]

Generate images of various characters in specified postures by providing the coordinates of each key point (major parts such as joints) as information that represents the posture to be taken by the person.

[Use Case UC-1B: Attribute Specification, Image]

Generate diverse and natural images by specifying the attributes of the target (e.g., gender, clothing color, etc.)

[Use Case UC-2A: Structure Sequence Specification, Video]

Generate natural videos of various characters moving in the specified postures, given a list of coordinates for each organ point, which is information on the postures to be taken by the person.

[Use Case UC-2B: End Image Specification, Video]

Generate natural and diverse videos in which a starting point image and an ending point image are given and "interpolated" between these images.

Input/output

The inputs for each use case are given below, except for UC-0, which has no constraints. The goal of the corresponding function in each use case is to produce natural and diverse output for what is not specified while following the specification of the input constraints. In addition, detailed information such as the frame rate of the video is omitted.

- Image
  - UC-1A: Coordinates of each organ point representing the structure (posture) (see Figure 5.8)
  - UC-1B: Attributes such as gender, clothing color, etc. specified from a list of options

- Video
  - UC-2A: Coordinates of each organ point representing the structure (posture), arranged as a list
  - UC-2B: Two images that serve as the start and end points of the animation, respectively

Fig. 5.8 generate image from organ point coordinates

5.2 Specific task

In the content generation system represented in the examples, the following points about quality are particularly unique. In this system, advanced functions that can be described as creative, such as "creating new images and videos," are realized. For this reason, the quality characteristics to be achieved are expressed in very abstract terms, such as "naturalness," "variety," and "maintaining the taste (style, etc.) of the input image. Therefore, it is ultimately necessary to have a human evaluator, but in reality, the question is what kind of automatic judgment criteria can be realized.

5.3 Expected quality characteristics

In the following, we will discuss the expected quality characteristics of image and video generation systems, referring to examples, which are mainly categorized as "adequacy" or "satisfaction" in ISO 25010 (SQuaRE).

5.3.1 Quality characteristics common to all use cases

[Quality characteristic QC01: Naturalness]

Each frame of the generated image or video is natural. For example, there is no sense of discomfort when compared to illustrations that already exist, or when said to have been drawn by a person.

[Quality characteristic QC02: Sharpness]

The generated image or video is clear. For example, even when viewed or printed at high resolution, there is no loss of shape, color, or noise.

[Quality Characteristics QC03: Diversity]

A variety of images or videos are generated. For example, the character's clothing and posture are not always the same. This also confirms that there is no technical issue (called Mode Collapse) that causes the generative model to approximate only a limited number of points in the distribution.

[Quality characteristic QC04: Social appropriateness]

The generated image or video is not socially inappropriate. For example, images that appear to be naked, or images that evoke a sense of cruelty and generate physiological disgust are not generated.

5.3.2 Quality characteristics for content specification

[Quality characteristic QC05: Conformity with the specified structure]

The generated image is consistent with the specified structure. As an example in Use Case UC-1A, if the right arm is supposed to be raised in the specified structure, the right arm is raised in the generated image. As another example, if the area placement of which object is placed at which position in the image is specified in the specification structure, the image is generated in a way that matches the specification.

[Quality characteristic QC06: Match with specified attribute]

The generated image matches the specified attributes. As an example in Use Case UC-1B, gender, clothing, facial expression, skin color, hairstyle, body shape, and accessories are specified as the designation. As another example, when a reference image indicating texture and style is given as a specified attribute, an image reflecting that specification is generated.

5.3.3 Quality characteristics related to video

[Quality characteristic QC07: Naturalness as video]

The generated video is natural. For example, compared to a video that already exists, there is no sense of discomfort as an image sequence or even if it is said to be a real video.

[Quality characteristic QC08: Smoothness of video]

The generated video is smooth. For example, there are no areas where there is no continuity in color or posture.

[Quality characteristic QC09: Naturalness as a structural sequence]

The structural changes of the objects depicted in the structure of the generated video are natural. For example, the sequence of body structures (organ points) related to each movement, such as the way of running and jumping, is more natural than existing videos, and there is no sense of discomfort even if it is said to be a real video.

5.4 Technical Approach for Quality Assessment and Assurance

For most of the quality characteristics listed in 4.3, there are no objective indicators that can directly evaluate the degree of fulfillment, and thus it is difficult to automate the evaluation of the degree of fulfillment. Since human inspection requires a large amount of man-hours, it is important to consider a method of quality evaluation and assurance that is technically and practically feasible and effective, even if only partially. This poses a major challenge at the research and development level. In the following, we will discuss how to consider quality assessment and assurance methods, including those for which similar examples have not yet been reported.

Quality assessment and assurance involves many activities on different artifacts, such as training

data, training algorithms, and the resulting trained models. For example, if we consider the case where training data is created by ordering illustrators, it is necessary to mention, for example, sharpness and diversity as specifications. On the other hand, even if such training data is used, there is no guarantee that the output from the resulting trained model will be clear and diverse. For this reason, in the following, we mainly discuss testing, i.e., methods to check whether the quality characteristics are met by inspecting a large number of outputs from the constructed content generation system.

Most of the quality evaluation methods listed below are approximate evaluations of quality characteristics that cannot be made exact. For this reason, some of the items evaluated as "poor quality" by the methods may actually have "no problem" or even "good quality" when viewed by a person. However, it is important to pursue tools that can find problems more efficiently than to have people check a small number of outputs obtained by random sampling. It is desirable to confirm the effectiveness of such a tool through experiments and hypothesis testing.

Among the target quality characteristics, for those that are not specific to the content generation system in the example but are highly general, evaluation methods have been established and may be available as libraries or software applications. In this case, it is conceivable that such existing methods and their implementations can be used to evaluate the quality characteristics.

5.4.1 Evaluation by Indicators

Evaluate the target quality characteristic by defining a quantitative indicator that approximates, if not represents, the quality characteristic.

[Example] quality characteristic QC01: naturalness / quality characteristic QC02: clarity / quality characteristic QC03: diversity

For the diversity of images and videos, it is sufficient to evaluate that they are clear and diverse for a large number of outputs from the content generation system.

This can be evaluated by using trained discriminative models for the images and videos to be generated, and by using the distribution of discrimination results and feature distributions. Inception score is a measure of the sharpness and diversity of a generative model, such as the inception score [Salimans+16] and Fréchet Inception Distance [Heusel+17]. The inception score uses a trained discriminative model for image classification, and is higher if each generated image has a distinct class in the discriminative model (peaks in only one class), or if the image as a whole is distributed over a wide range of classes. The Fréchet Inception Distance is the distance between data sets calculated using the intermediate feature distribution calculated from the trained discriminative model. The distance between the feature distributions of the real data (training data) set and the generated data set is calculated to evaluate whether the generated data is as clear and diverse as the training data. For video evaluation as well as image evaluation, a learned discriminative model for the Action Recognition task, which outputs action types based on video input, is used, and an equivalent distance calculation is also performed.

For each individual industrial application, these schemes are applied. Learn and utilize discriminative models for variables that control the features of images and videos. For example, for an illustration image or video that we want to generate and control, we learn a discriminative model to determine its type. By calculating the inception score using the learned model, it is possible to evaluate the generation of various images and videos in the control target. When training the discriminative model, the selection of training data from the entire training data, partial selection, or selection of N types, can be used to improve the index evaluation through output evaluation by ensemble of models. As the training data for the discriminative model, we use the constructed domain image to be generated. For example, it can be

composed of images drawn by illustrators. Alternatively, it can be compared with illustrations that can be naturally collected on the Web.

[Example] Quality Characteristics QC08: Smoothness of Video

The system calculates the statistics of the optical flow between images and evaluates the amount that exceeds the threshold of the amount of change that is natural for a video. By checking whether the amount of change in the video output by the system exceeds the threshold, videos with unnatural smoothness can be detected.

[Example] Quality characteristic QC09: Naturalness as a structural sequence

The amount of change in the relative distance between organ points connected in the body structure is quantified, and the amount that exceeds the threshold of natural change as a structural sequence is evaluated. By checking whether the amount of change in the structural sequence in the video output by the system exceeds the threshold, videos with unnatural structural sequence can be detected.

5.4.2 Building quality assessment AI by machine learning

Although naturalness and the like are criteria that cannot be put into words (and cannot be implemented in software using deductive rules), it is possible to implement quality assessment AI for output results if training data is available.

In cases where existing learned discriminative models cannot be used due to domain differences, we will build our own quality assessment AI. For example, the pose estimation of a person differs greatly between the pose estimation of a photorealistic image and that of an illustrative image. It is conceivable to build a quality evaluation AI by utilizing the training data used to build the original content generation system. If we are aware of and pre-label posture and attribute evaluation, we can build an AI that estimates posture and attributes from images and videos and evaluate their quality.

[Example] quality characteristic QC01: naturalness / quality characteristic QC02: clarity

The data collected when building a content generation system is supposed to represent naturally occurring images and videos. In the generation model of another architecture, we will use some of the collected data to create a discriminator that can determine whether an image is natural or not.

For example, as an AI to evaluate the naturalness of an image, we can use GANs training to build a discriminator to determine whether the data is real or not, and use the score of the discriminator to evaluate the naturalness and sharpness of the image. Alternatively, a model that has learned image restoration can be used to encode and decode images into image latent vectors, and the restoration score of the generated data can be used to evaluate the naturalness and sharpness of the image.

In addition, the illustrations that are the subject of this project can be collected from the Web. By adding collapses and noise of a size that would obviously be problematic, we can create data that represents unnatural images and videos. Using these two types of data, natural and unnatural, it is possible to build a quality assessment AI that can detect the assumed unnaturalness. Similarly, it is also possible to detect unclear images by building a model that classifies clear image data as training data by creating image data with shape and color corruption added as noise or reduced resolution to clear training data.

[Example] Quality characteristic QC04: social appropriateness

Build an AI that identifies socially inappropriate data. It can be used to eliminate inappropriate images and videos from training data or to detect inappropriate ones from a large number of outputs of

a content generation system. Note that similar techniques have been established for search engines and other applications.

[Example] Quality characteristic QC05: Match with specified structure / Quality characteristic QC06: Match with specified attribute

There has been a lot of research and development in AI for estimating the pose of a person in an image or video [Sun+19, Pavllo+19,Kocabas+19]. In Use Cases UC-1A/2A, we are considering using such techniques to evaluate the output from a content generation system. Specifically, the degree of conformity with the specified posture can be evaluated by estimating the posture in the output and measuring the distance between it and the posture indicated by the input.

[Example] Quality characteristic QC07: Naturalness of video

Similar to QC01, create a discriminator [Wang+18] such that when a few images are extracted from a video, we can determine whether they are natural as image examples.

[Example] Quality characteristic QC08: smoothness of video

Create a discriminator to determine whether the distribution of optical flow among images is natural or not when several images are extracted from a video, as in QC01.

[Example] Quality characteristic QC09: Naturalness as a structural sequences

Utilize the same discriminant model as in QC01 to determine whether the structural sequence is natural or not [Cai+18,Barsoum+18,Kundu+19]. When a few images are extracted from a video, we create a discriminator that determines whether the structural sequence extracted from the images is natural or not.

5.4.3 Comparison with other implementations such as rule-based AI

If we can obtain implementations of the same function or the target quality evaluation even if their performance is slightly worse, we can use them to perform pseudo-quality evaluation (pseudo-oracle).

[Example] Quality characteristic QC06: match with specified attribute

For simple attributes such as the color of the clothes, a simple implementation can be done, such as extracting the main colors from the image by using an image processing library. This may not necessarily match the actual "color of the clothes", but by comparing with such a simple implementation, the inspection can be implemented.

5.5 Quality Assurance Level

The quality characteristics listed so far in this chapter relate to the output of the entire system, which is directly experienced by the user, and will be displayed to the user almost directly as the output from the model. This is because it is difficult to perform post-processing such as filtering of images and videos by rule-based programs. For this reason, the quality characteristics listed in this chapter are for model robustness, which is one of the quality assurance levels in Chapter 2.

Chapter 2 also states that the quality assurance of data, models, systems, and processes must be appropriate to meet customer expectations. In this example, for example, quality assurance levels

corresponding to different usage situations can be considered as follows

- Level 1: A level that should be guaranteed for use by people who know the internal behavior of the system. For example, an organization that operates a video production service and includes the development team of the content generation system.
- Level 2: The level that should be guaranteed for use by outsiders who do not know the internal behavior of the system. For example, when a content generation system is delivered to an animation production company for use.
- Level 3: The level that should be guaranteed when the system is used by a wide variety of users, including those with malicious intent. Level 3: A level that must be guaranteed when a content generation system is released on the Web as a service that can be used by an unspecified number of users.

The difference between the requirements for Level 1 and Level 2 is the orientation of stability, where each output is stable and "good". In the case of internal use, such as Level 1, the output is manually checked before it is sent to the user, and it is easy to select the best output from multiple outputs or to re-send the output. For this reason, when evaluating each quality characteristic, it is possible to evaluate only the best of multiple outputs (maximum value, etc.) and not pay much attention to quality variations (variance value, etc.) or the worst case. In the case of Level 2 and above, where the output is used by external users, stability is more important, i.e., each output should be stable and "good" or at least not "bad".

The difference in thinking between Level 2 and Level 3 is the ease of defining, limiting, and agreeing on specifications, especially the scope of application. For example, when images are input in use case UC-2B, the scope of application may be defined as only images by a specific illustrator can be input. Then, quality assurance can be considered only under this assumption. Level 3 is a B2C assumption, and from the viewpoint of the attractiveness of the service, it is difficult to limit the scope of application, and even if a detailed scope of application is defined, it is assumed that users may use the service without reading it carefully or thoroughly and become dissatisfied. For this reason, it is necessary to conduct quality assurance activities assuming a wider variety of inputs.

Table 4.1 shows the process of how to implement the evaluation activities for each of the quality characteristics illustrated in Section 4.1.4 according to these levels. In this section, the evaluation activities associated with the design and construction of the model are divided into Phase 1 and those for quality assurance into Phase 2. The roles of each are as follows.

In Phase 1, evaluation activities are conducted during the design and construction of the model. In other words, quality characteristics are evaluated as an objective function for building models through training and optimization, or as a criterion for evaluation and improvement of architectures, training methods, etc. For example, QC01-03 (naturalness/vividness/variety) is an important quality characteristic for all use cases. In addition, the inception score and other evaluation metrics for QC01-03 (4.1.4.1) are generally used as evaluation metrics for content generation by generative models. Therefore, the activity of "evaluation by QC01-03 metrics" can be positioned as one in which the developer who designs and builds the model checks the quality of the model and iterates design improvement, training, and optimization until a certain level of quality is achieved.

In Phase 2, after the model has been built to a certain extent, activities are conducted to further check and improve the quality of the model. In other words, the evaluation is conducted from a different perspective than that of the developer who designed and built the model. For this purpose, not only the quality characteristics that are the essential goals of the model, but also additional quality characteristics are evaluated, and AI for evaluation and different evaluation data sets are prepared for evaluation. For

Table 5.1 Process and level of quality assurance in a content generation system

	Guiding Principles	Phase 1: Quality to drive the build (design, training, optimization)	Phase 2: Quality to evaluate from different perspectives and data, quality to build and evaluate additional mechanisms (including QA and third parties)
Level 1	Evaluate for the possibility of selection from multiple outputs	Basic evaluation in driving improvement of the generative model (evaluation of indicators in QC01-03, 07-08).	Evaluation for additional quality characteristics (QC04).
Level 2	evaluation with emphasis on stability	evaluation of core functions for use case realization (evaluation using existing AI in QC05, 06).	Evaluate the quality metrics addressed in Phase 1 using additional domain-specific datasets and
Level 3	Evaluate assuming a variety of inputs	Evaluate using common metrics and AI implementations that are available.	Additional evaluations using evaluation tool implementations.

example, for QC01-03 (naturalness/vividness/variety), an evaluation dataset can be constructed separately from the training dataset or an original evaluation AI can be constructed (4.1.4.2) to reflect the perspectives specific to the target domain. Specifically, efforts should be made to take into account domain-specific points such as the appropriate way to draw lines in illustrations, whether the image or video contains only people or people and background, and how much diversity to assume in people's clothing.

As described above, Phase 1 deals with quality characteristics that are fundamental to the function in the use case and therefore used to drive the model design and construction, and that can be evaluated using general metrics and AI. Phase 2 deals with additional quality characteristics, quality characteristics that require the construction of specific metrics or AI.

5.6 Test design examples

This section presents a case study in which the test design corresponding to the quality assessment described so far is applied to an image generation system. As the target system, SPADE [Park+19][*1]is used.

5.6.1 Outline of the target system

SPADE receives the following two inputs and generates images according to the inputs.

label map A representation of the correspondence between a region and the objects in that region, such as "sky" for this part of the image, "tree" for this part, etc., which specifies the placement of

[*1] https://nvlabs.github.io/SPADE/

the objects in the generated image.

Style Guide Image　Specifies the style of the generated image by using a reference image. For example, if a sunset image is specified, the generated image will show a red sun low in the sky, and trees and the ground will appear dark and shadowy.

Figure 5.9 shows an example of the output from SPADE. The columns correspond to the label maps and the rows to the style guide images, which are combined to produce different images.

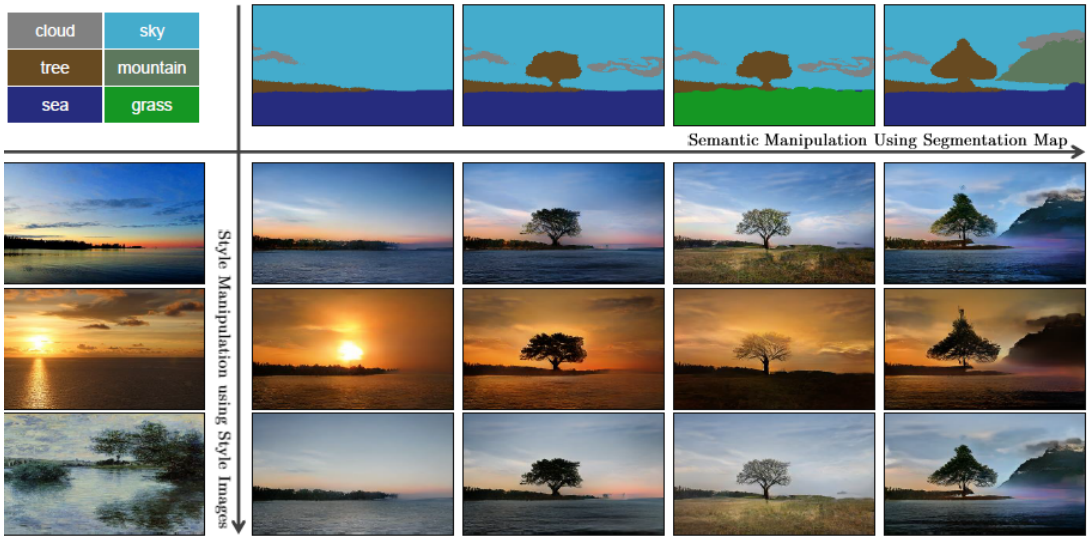


Fig. 5.9　Example of SPADE output [Park+19]

5.6.2　Quality characteristics to be covered

In the 5.3 section, we discussed quality characteristics for systems that generate images and videos, and many of these quality characteristics, including QC01 (naturalness), are general ones that are common to all image and video generation systems. For QC05 (conformance with specified structure) and QC6 (conformance with specified attributes), it is necessary to specify what is meant by "structure" and "attributes" for each target system.

In the example of test design for SPADE, we first deal with QC01 (naturalness) and QC02 (clarity) as the same as those defined in the 5.3 clause. Next, the following quality characteristics are specific to SPADE.

QC05' (match with specified label map)　The object placement specified in the label map is observed.

QC06' (match with specified style guide image)　The style specified in the style guide image is reflected.

Here, a specific question for QC01 (naturalness) and QC02 (sharpness) is whether high quality can be stably achieved for a variety of possible inputs consisting of label maps and style guide images. For example, let us consider the use of SPADE to generate images of driving scenes to train or test AI for automated driving. In this case, by specifying a label map as input, we can generate a set of images that systematically cover the possible positions of a pedestrian. Similarly, we can generate a set of

images with different brightness and color for various weather conditions and time of day by using style guide images. Given these use cases, it is important to test SPADE systematically by changing the input exhaustively based on certain criteria. In addition, systematic testing by varying the input is effective because it cannot be denied that a small change in the input may cause a large change in the output, as has been widely discussed as a hostile sample for discriminative models.

5.6.3 Test-design

An example architecture for testing SPADE is shown in Figure 5.10. The output of SPADE is evaluated.

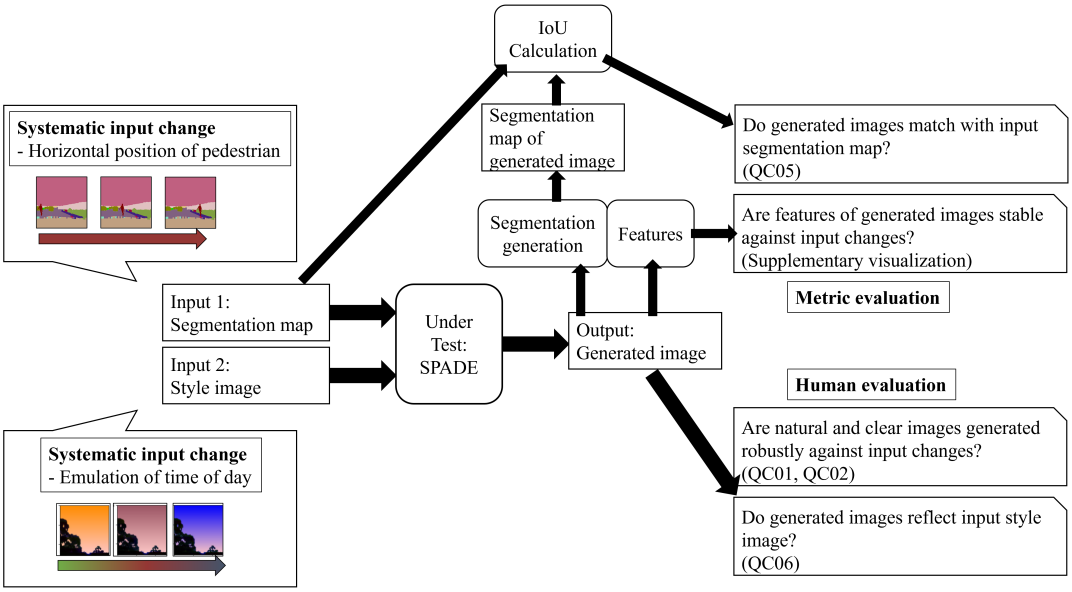


Fig. 5.10 Architectural design for testing against SPADE

A sequence of images generated for a set of inputs is evaluated in terms of QC01 (naturalness) and QC02 (sharpness). For QC05' (matching with the specified label map), we generate a label map from the images generated by SPADE and compare it with the label map of the input. The IoU index evaluates the degree of overlap of the regions for each class on the two label maps. This comparison method for QC05' is used not only in SPADE but also in the semantic segmentation tool itself. For QC06' (matching with the specified style guide image), we assume visual confirmation, although we can consider automating the process to determine the style match.

Among the test inputs, it is not possible to generate a large number of label maps if they are drawn by hand. It would be more practical to obtain the original label maps by either using a dataset that already contains label maps or by generating label maps from existing images, and then systematically processing them (e.g., changing the positions of pedestrians). For style-guided images, we would classify existing images as morning glow, backlighting, cloudy, etc. (attributes that are often discussed in the automated driving domain). By artificially modifying the color and brightness of the sky, it is possible to construct a sequence of images that gradually darkens and becomes night.

5.6.4 Experiment

We tested SPADE according to the design described in 5.6.3subsection , the version of SPADE that was released in December 2020 (version of October 18, 2019 1a687ba) [*2] was used. For the implementation of the semantic segmentation method, we used the DeepLab v2 trained model [*3].

In the following, we describe the results of the test runs for the two input variants. We did not perform a complete evaluation of the quality of SPADE from all perspectives as described above, but only present a few cases where we confirmed the effectiveness of the tests.

First, an example of the implementation results for the case where the pedestrian position is systematically shifted sideways in the input label map is shown in Figure 5.11. In the upper part of the figure, the positions of the pedestrians are systematically shifted to the left and right in the input label map. The lower part of the figure shows the images generated by SPADE for these inputs. It can be said that the generated images maintain a certain level of naturalness (QC01) and conformity to the specified structure (QC05) because the pedestrians can be sufficiently recognized regardless of their positions. Although there are some changes in the pedestrian's clothing, they are within the natural range. We also measured the IoU index, and found that it did not change significantly depending on the position of the object, but remained stable and constant. We also measured the IoU index and found that it did not change significantly depending on the position of the object. These results show a certain robustness of the input image with respect to the position of the pedestrian. In reality, we will have to perform similar evaluations on a large number of images, but we will not go that far in this trial. We also tried to move the position of the bicycles on the roadway up and down in the image (to the back or to the front as seen from the car), but the results were similar, so we omit the details.

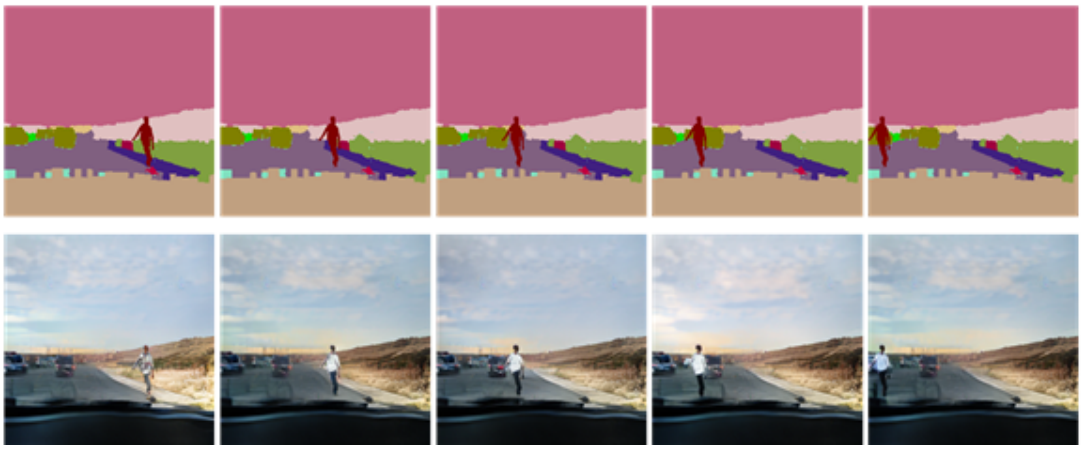


Fig. 5.11 Example of test results for SPADE (bottom image generated for systematic changes in the label map in the top row)

Second, we show the evaluation of the case where the time of day changes and the color of the sky changes in the input style guide image. In this style guide image, the sky was artificially colored from the original image, and the color was changed from day-like to night-like. Figure **??** shows an example of the change in the artificially applied style guide image and the corresponding change in the output

[*2] urlhttps://github.com/nvlabs/spade/
[*3] `https://github.com/open-cv/deeplab-v2`

image. In the output image, the position of the sun changes to reflect the change in the time of day. As a result of systematically changing the style guide image in this way, the generated image was sometimes significantly corrupted near a certain shade. The style guide image and the generated image in this area are shown in Figure 5.13. Even though the change in the style guide image is so slight that it cannot be seen by the human eye, the generated image collapses in the fourth and sixth rows from the left. This unstable behavior continues in the future. Therefore, it can be said that all four quality characteristics (QC01, QC02, QC05, and QC06) addressed in this test design case are not satisfied. In addition, even by monitoring the feature values of the images, the feature values were clearly unstable and changing near the corresponding style guide images (confirmed by the three indicators of SIFT, AKAZE, and ORB implemented in the OpenCV library). After systematic testing, we were able to detect cases that were not robust to changes in the style guide image.

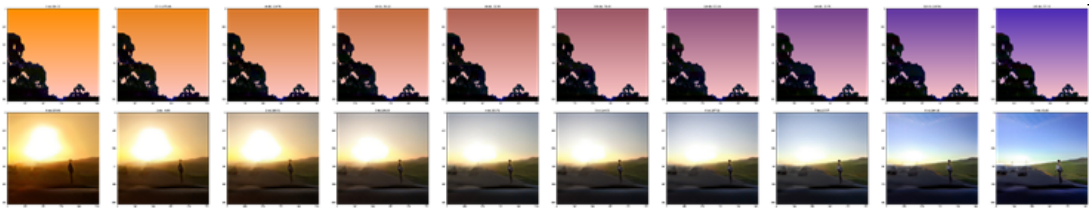


Fig. 5.12 Example of SPADE output changes for a style guide image (the bottom row is the image generated for the systematic change of the style guide image in the top row)

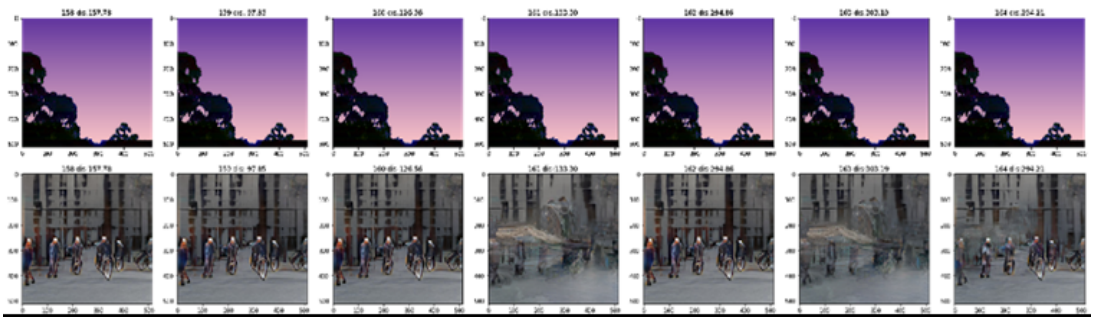


Fig. 5.13 Example of test results for SPADE (the bottom image is generated for the systematic change of the style guide image in the top row)

5.6.5 summary of test design examples

In this section, we have presented a systematic test case study for SPADE. When the input is data with complex information such as images or label maps, it is often difficult to collect existing data to systematically cover a certain viewpoint. For this reason, we have been processing the existing data. In some cases, systematic testing confirms the robustness of the data, while in other cases, specific problems are found. How to fix problems is a major issue not only in GAN-based generative models but also in deep learning in general, and we think it is important to understand specific problems and their trends. In this case study, we were able to show an actual example of this.

Acknowledgements

The implementation results shown in 5.6section are the results of our work in the Practical Software Development Exercise in the Top SE Program at the National Institute of Informatics. We would like to thank Hisanori Iijima (Fujitsu Limited), Hiroyuki Oikawa (Toshiba Digital Solutions Co., Ltd.), Takara Kasai (Sony Corporation), Tao Komikado (Fujitsu Laboratories Ltd.), Ryuji Go (NTT Data Corporation), and Sho Takano (Denso Corporation) for working on exercises that conform to these guidelines and providing us with the results.

5.7 References

[Goodfellow+14] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In NIPS 2014.

[Karras+18] Tero Karras, Timo Aila, Samuli Laine, Jaakko Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In ICLR 2018.

[Brock+19] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In ICLR 2019.

[Karras+19] Tero Karras, Samuli Laine, and Timo Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. In CVPR 2019.

[Hamada+18] Koichi Hamada, Kentaro Tachibana, Tianqi Li, Hiroto Honda, and Yusuke Uchida. Full-body High-resolution Anime Generation with Progressive Structure-conditional Generative Adversarial Networks. In ECCV Workshop 2018.

[Hamada+19] 濱田晃一 , 李天琦. AI によるアニメ生成の挑戦. In DeNA TechCon 2019. https://www.slideshare.net/hamadakoichi/anime-generation

[Hamada+20] Anime Generation with AI. DeNA. 2020. https://www.slideshare.net/hamadakoichi/anime-generation-ai (Generated Anime: https://youtu.be/X9j1fwexK2c)

[Salimans+16] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen. Improved Techniques for Training GANs. In NIPS 2016.

[Heusel+17] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Sepp Hochreiter. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In NIPS 2017.

[Russakovsky+15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, and Michael Bernstein. ImageNet large scale visual recognition challenge. In IJCV 2015.

[Odena+17] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional Image Synthesis with Auxiliary Classifier GANs. In ICML 2017.

[Miyato+18] Takeru Miyato and Masanori Koyama. cGANs with projection discriminator. In ICLR 2018.

[Zhang+18] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In ICML 2018.

[Wang+18] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, Bryan Catanzaro. Video-to-Video Synthesis. In NeurIPS 2018.

[Cai+18] Haoye Cai, Chunyan Bai, Yu-Wing Tai, Chi-Keung Tang. Deep Video Generation, Prediction and Completion of Human Action Sequences. In ECCV 2018.

[Barsoum+18] Emad Barsoum, John Kender, Zicheng Liu. HP-GAN: Probabilistic 3D human motion prediction via GAN. In CVPR 2018.

[Kundu+19] Jogendra Nath Kundu, Maharshi Gor, R. Venkatesh Babu. BiHMP-GAN: Bidirectional 3D Human Motion Prediction GAN. In AAAI 2019.

[Sun+19] Ke Sun, Bin Xiao, Dong Liu, Jingdong Wang. Deep High-Resolution Representation Learning for Human Pose Estimation. In CVPR 2019.

[Kocabas+19] Muhammed Kocabas, Salih Karagoz, Emre Akbas. Self-Supervised Learning of 3D Human Pose using Multi-view Geometry. In CVPR 2019.

[Pavllo+19] Dario Pavllo, Christoph Feichtenhofer, David Grangier, Michael Auli. 3D human pose estimation in video with temporal convolutions and semi-supervised training. In CVPR 2019.

[Park+19] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic Image Synthesis with Spatially-Adaptive Normalization. In CVPR 2019

6. Voice User Interface (VUI)

6.1 Assumed System

In this area, we envision a so-called "Voice User Interface (VUI)", which is a system that listens to the speaker's words, interprets the intention of the listened text, and executes the actions intended by the speaker.

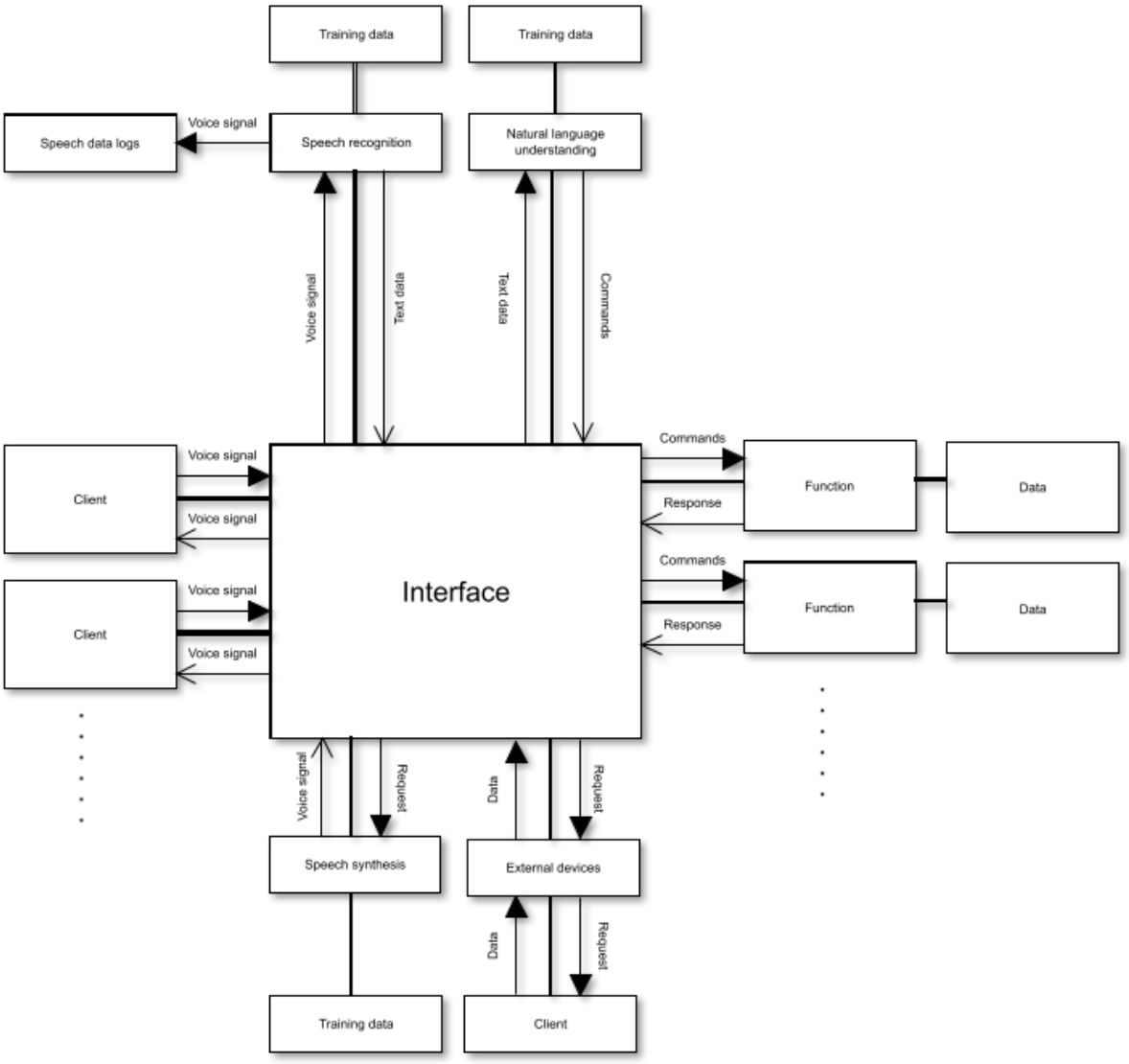
We assume that the system uses the following three types of machine learning.

- Machine learning used for speech recognition, which converts the speech signal input from a microphone into text.
- Machine learning used for "natural language understanding," which interprets what the converted text is intended to be and converts it into commands to perform functional actions.
- Machine learning used in text-to-speech, which converts text into speech signals

The flow of the system operation is assumed to be as follows The overall picture is shown in Figure 6.1.

1. Input audio signal to the system
2. The input speech signal is passed to the speech recognizer and output as text.
3. Text is passed to the natural language understanding function and output as a command to activate the desired function
4. Input command to the function corresponding to the command (e.g. weather acquisition function, music acquisition function, etc.)
5. Data is processed by the corresponding function
6. Outputs a response from the corresponding function
7. Input the response from the relevant function
   a. in the case of input to the text-to-speech function
      i. Input the response from the relevant function to the text-to-speech function
      ii. Output as a speech signal from the text-to-speech function
   b. For input to an external device
      i. Enter the response from the corresponding function to the external device.
      ii. Processing the response by the external device
      iii. Execute the external device function

Fig. 6.1 Assumed system operation flow



6.2 Features of the VUI System

6.2.1 Voice Recognition

For the speech recognition part, it is required to be able to "correctly interpret the same voice input string as a character string. For example, the same voice input "What is today's weather? the input voice signal will change depending on the following conditions.

- Gender (male, female, etc.)
- Age (high/low)
- Tone (accent, fast speech, voice color, Kansai/Tokyo intonation)

- Position of sentence breaks ("What's the weather? /"What's...the weather? etc.)
- emotion (e.g., "kind/severe")
- Pronunciation by non-native language learners

In addition, for speech recognition, "the system must operate correctly if the system user environment is within the product warranty range" is also required. The following conditions are possible.

- environment
  - audio environment
    - audio noise (e.g. TV, talking voice)
    - Noise noise (driving noise, household noise, etc.)
  - Installation environment
    - Vibration (e.g. shaking due to unstable footholds)
    - Closure (e.g., effects of reverberation, echoes from windows)
- use case
  - distance of speech

In some cases, it is required to recognize the voice of a person other than a specific person for security reasons. In that case, the following conditions are considered.

- speaker recognition
  - identification of an individual

6.2.2 Natural language understanding

For the natural language comprehension part, "the ability to understand different expressions as standard sentences" is required. Here, we consider Japanese. The following conditions are considered.

- tone of voice (honorific, imperative, youthful, etc.)
- postpositional particle (e.g. variations in "te-ni-oha")
- grammar (e.g., word order changes, ending a sentence with a noun)
- Abbreviations (abbreviations such as "koibana", abbreviations of singers' names, abbreviations of place names, etc.)
- homonyms (e.g. "rain/ame", "want to see/look", homophonic place names)
- Proper nouns (e.g. place names, names of places)
- Japanese English (e.g. laptop/ノートパソコン, electrical outlet/コンセント)
- Popular words (e.g. baeru, moyaru)
- Dialects (Kansai dialect, Tsugaru dialect, etc.)

6.2.3 Speech synthesis

Speech synthesis is required to "convey voice messages that can be understood by users of the system". Here, we consider the Japanese language. The following conditions are considered.

- Pronunciation (Kanji reading. "tsukitachi/ichinichi", idioms, famous people, etc.)
- tone (accent, emotion, tone of voice, etc.)
- Speech patterns (e.g. sentence breaks)
- information (speed, "too long/too short", etc.)
- Voice quality (male, female, voice actor, etc.)

6.2.4 Other - infotainment

Infotainment is a term coined from the combination of information and entertainment, and is an element or system that provides information and entertainment.

In addition to performing functions, VUI systems are often required to provide information, conversation, and other forms of infotainment. Therefore, non-functional characteristics such as diversity of conversation and interestingness are also mentioned.

6.3 Specific Issues

6.3.1 System issues

A common feature of VUI systems is the problem that, since voice input is the only entry point, if voice recognition fails, the natural language understanding function also fails to function properly, directly leading to unintended output. This problem can be dealt with, for example, even if speech recognition fails and words with similar but different nuances are input to the natural language comprehension function, if the failure is predictable to some extent, the natural language comprehension function can handle it.

The following is a list of issues using smart speakers and car navigation systems as examples of systems using VUI.

One of the major features of smart speaker products is that many independent functions such as weather, music, and fortune-telling can be executed for "one entrance" of voice input only. This is a feature of smart speakers, but the wide variety of functions also makes it difficult to identify target users. The difficulty in identifying target users is related to the amount of training data for the voice recognition function and its selection.

In addition, when adding or changing functions, the probability of misrecognition may increase due to the similarity of voice commands with existing functions. When adding or changing a feature, it is necessary to consider the impact on the whole system.

In the case of a car navigation system, it is mainly used in a vehicle. In the case of a running vehicle, the noise during driving affects the speech recognition. Therefore, preprocessing to discriminate between speech and noise is important.

In addition, since the vehicle is a closed space, speech echoes are generated. As a result, it is difficult to determine where the voice originates from, for example, whether the voice is coming from the driver or a child in the back seat. This makes the process of determining "which voice command should be prioritized" important.

Considering the fact that the vehicle is in motion, a miscommunication on the part of the speaker or on the part of the speech synthesizer may lead to an accident. In the case of miscommunication by the speaker, there is a possibility that the miscommunication will be interpreted correctly and the driver will be led to a place different from the intention. For example, if the name of a place "Hirosaki" is read out as "Hiromae," the driver may be distracted or turn his or her attention to the screen, which may lead to an accident.

The fact that the vehicle is moving is another major feature of the car navigation system. For example, if a driver asks for the location of a nearby convenience store, it is difficult for the driver to stop even if the system outputs the location of the convenience store a few meters away because the vehicle is moving. Of course, even if it is close, it is difficult to go to a convenience store in the opposite direction.

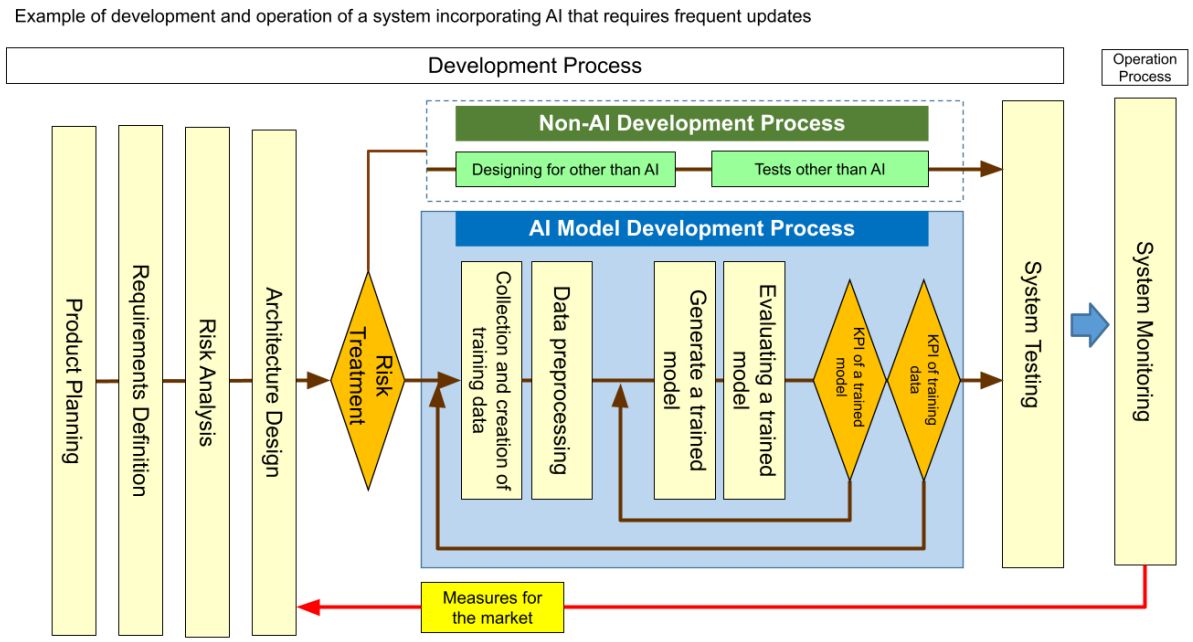
However, if the convenience store ahead is far away, it may be better to make a U-turn. It is necessary to deal with how to process "near" in the system side after natural language understanding.

6.3.2 Process Agility Issues

In smart speakers, it can be frequent to respond to new words if they are introduced, or to commonly used words as a result of users starting to use them. This is a unique problem because, unlike in the case of automobiles, frequent updates are required. It is considered necessary to have a development and operation process to cope with such updates.

.

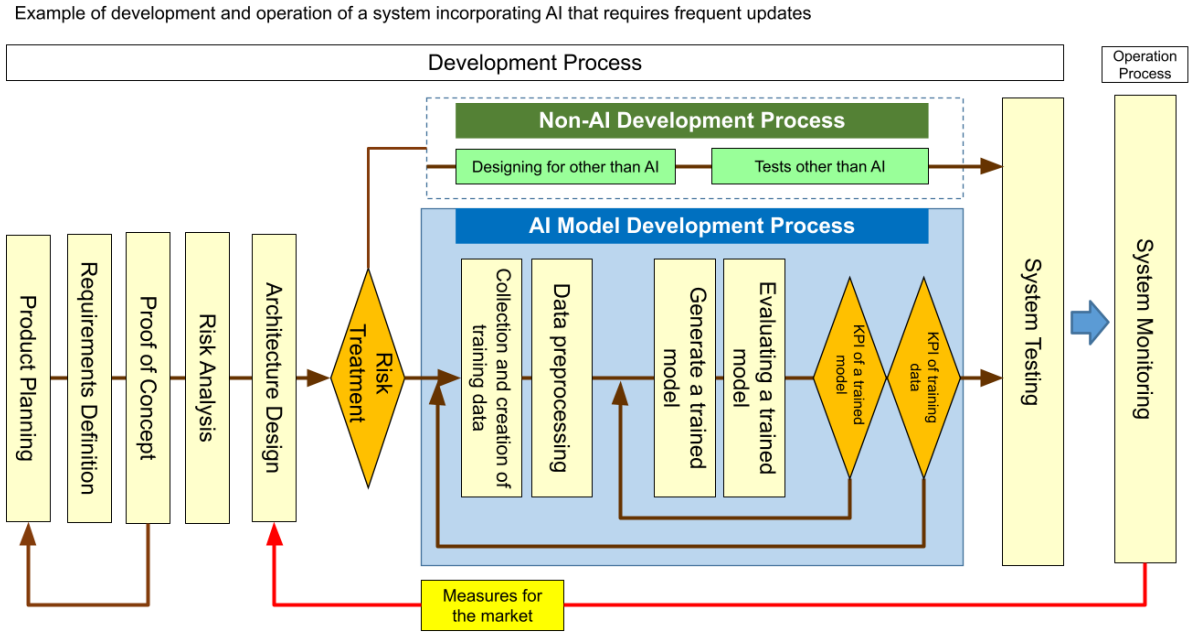
Fig. 6.2 Example of developing and operating a system incorporating AI that requires frequent updates



One example of achieving Process Agility is Figure 6.2. After the release, it is essential to have an operational system to catch up with the needs and problems of the language through monitoring and surveillance. After the release, it is essential to have an operational system to catch up with the needs and problems of the language by monitoring and surveillance. From there, those problems are picked up, reflected in the model, and released. It is necessary to have a system that can create a cycle of reflection and monitoring in the form of monitoring and surveillance.

Unlike smart speakers, it is difficult to update the car navigation system frequently. In the case of such a system, PoC verification must be conducted at the initial stage, and the way it is used in the market and speech patterns must be thoroughly examined. After that, the system is developed and released. Even in such a system, post-release monitoring and surveillance is conducted, and catching up with deviations from the assumed speech patterns occurs. It is necessary to build an operational system that picks up these problems, reflects them in the model, and updates them. An example of achieving Process Agility in this system is shown in Figure 9.2.

Fig. 6.3 Example of development and operation of a system incorporating AI that cannot be updated frequently



6.3.3 Safety issues

VUI-based systems are easy to operate because they work by talking to the user. Although it is convenient and easy to operate, it is necessary to pay sufficient attention to safety.

Some smart speakers are equipped with a function that allows users to purchase products simply by speaking. Although this function makes it easy to purchase products, misspoken words or malfunctions in voice recognition may result in the purchase of products or quantities that do not match the intention. There are also cases where a product is purchased in response to a commercial voice. In the case of such a function, it is necessary to increase the number of purchase steps and to enhance the confirmation function, so that it can work easily and does not disadvantage the user.

In addition, the connection of smart speakers to external devices is a very important function for convenience. However, when dealing with external devices, especially products such as air conditioners that require stringent checks against quality incidents, the establishment of agreements and guidelines on the part of the smart speaker and the manufacturer of the external device to which it is connected can be cited as a quality standard to prevent unexpected incidents.

In the case of a car navigation system, it should be noted that it is mainly used while driving. For example, when making a right turn, even if the information is given just before the intersection, there are many cases where the driver cannot enter the right turn lane, and there is a possibility that an accident will occur if the driver suddenly changes lanes in response to the instruction. The amount of information conveyed is also important. If a large amount of information is given at once while driving, it will interfere with the driver's concentration.

The quality standard is to prioritize safety when unintended speech occurs at unintended times.

6.3.4 Privacy Issues

VUI-equipped systems, especially smart speakers, are often connected to functions that handle personal information, such as e-mail functions and SNS connections. In this case, it must be possible for the user himself to handle the system, but not for others to use it. To achieve this, we will consider introducing a system to protect information that can only be accessed by the individual, such as a voiceprint authentication system to identify the individual.

6.4 Expected quality

The expected quality common to all VUI systems is that "speech input is recognized as intended by the speaker and the intended function is executed".

In order to achieve this, the Data Integrity perspective is to have the same behavior with various voices. The various voices are the pronunciations of different genders, different ages, different tones, different emotions, and different language learners as mentioned in the speech recognition function of 6.2.1. In order to perform the same movement, it is necessary to convert the text into an appropriate text when the text is passed from the speech recognition function to the natural language understanding function. Therefore, it would be a good idea to determine the necessary elements of the product from the previously mentioned elements, and set acceptance criteria for each of them.

Similarly, the same behavior in various expressions is also mentioned in terms of data integrity. The various expressions are the elements of tone, particle, grammar, abbreviation, homonym, Japanese English, and popular words as mentioned in the natural language understanding function of 9.3.4. Since the number of combinations of language expressions is explosive, it is better to decide the range to guarantee the recognition and to ensure the quality within the range.

As a requirement for outputting the result, it is also important to be able to convey a voice message without misreading as a viewpoint of data integrity. These are the elements of pronunciation, tone, speech, and information mentioned in the text-to-speech function of 9.4.2. In terms of pronunciation, it is not realistic to cover all words. Therefore, it is better to limit the range to be guaranteed from the viewpoint of how the product is used and to ensure the quality within the range.

Also, new words are born every day. From the perspective of Model Robustness, it is also necessary to create a system that can quickly catch up with these new words and quickly respond to and update them.

From the perspective of system quality, it is important to consider what kind of situations the system including VUI will be used in, and to consider the quality of how it will be used in those situations. For example, in the case of smart speakers, since they are placed in the home, they are likely to be used in situations where people are living at home. One of the requirements is that noise should not interfere with recognition. Depending on the location where the product is installed, the TV or other people's voices may enter. In the case of a car navigation system, it is mainly used while driving. It should not be affected by noise and vibration during driving or echoes in a small space. It is a good idea to pick up the elements that require products for the elements of voice noise, noise noise, shaking due to location, and reverberation listed in the system environment of the speech recognition function in 6.2.1, and establish acceptance criteria and confirm them.

In VUI, Customer Expectation is greatly influenced by "Voice". In the case of VUI, the user interface is not a screen but a voice. In the case of VUI, the user interface is not a screen but a voice, and the expectations for this voice vary depending on the target audience and product image. Do you prefer a

masculine, neutral, or feminine voice, a speech style that does not include emotion like a newscaster, or a speech style with ups and downs of joy, anger, sorrow, and pleasure like a drama or animation? These factors are also subject to quality assurance to ensure that the voice does not deviate from the expectations of the target audience or the product image. Since it is necessary to collect teacher data for speech synthesis, it is desirable to confirm whether the voice is suitable for the product image or not at an early stage such as from the specification stage to the early development stage.

Customer Expectation for smart speakers has a wide range due to the fact that the target user is not clearly defined. Therefore, it is necessary to define the target users for each function, such as weather and music, and consider the criteria for satisfying Customer Expectation. In Data Integrity, Model Robustness, and System Quality, it is necessary to consider the characteristics of smart speakers.

Customer Expectation in car navigation systems is one of the requirements for navigation without inadvertently diverting the driver's attention while driving. For example, it is desirable that the recommendation for entering a different lane in two lanes is made at an early timing or while the car is stopped at a red light, so that the navigation is done in a natural way like human navigation and does not interfere with the driver's attention while driving.

6.5 Test architecture

An overall example test architecture for software testing of VUI is shown in table6.1 (hardware testing is omitted). It would be appreciated if you could refer to it when you add functions.

Even though the system encompasses machine learning, it is still the same logical system as before except for the machine learning module. The overall testing suggestion here is "don't be confused by the term "machine learning", and test in stages with different granularity as before. Step-by-step testing at each test level (component testing, integration testing, and system testing) will make it easier to isolate problems in later processes.

The system testing methodology is proposed in 6.6.

In addition, usability tests targeting actual user groups after the release of the product are also included in the table because they are considered to be useful for improving the performance of the product in the future.

Table 6.1: Testing Architecture

Test Level	Test Target	Test Content	Description
component testing	Various functions (weather, open window etc.) System part except for machine learning part	functional test for each component	System components other than machine learning and various functions (weather, open window, etc.) can be tested in the same manner as before. It is recommended that the parts of the system that can be tested as components be tested to confirm that they have been tested. This will make it easier to isolate and resolve any issues that may arise after the machine learning part is connected.

	Speech recognition part Natural language understanding part Speech synthesis part	Performance test for training data and models	Each machine learning part needs to be checked for accuracy on its own. Examples of patterns required for learning are listed in Table 6.2. These patterns can be used as reference data or as individual features to evaluate their accuracy.
	Each component	Testing of the structure for each component	Focusing on the internal structure of each component, we will check if the components are built as designed.
		Verification test for each component.	After each component has been modified, check that the modified part has been properly .
Integration testing	Various APIs	API response, DB and modules functional testing after connection	The system is made up of multiple APIs that pass data to and from each other. Before starting the system testing phase, it is better to prepare the data in the expected patterns and check the APIs'responses in the integration testing phase. Otherwise, it will take a long time to isolate the problem in the system testing phase.
		Performance testing of APIs, servers, etc.	Can it handle the data under load? Check if the server scale-up and scale-out functions efficiently and if the response time is within the acceptable range.
		Confirmation test and regression test after modification	When changes or corrections are made, a regression test will be conducted to confirm that the changes or corrections have been made correctly and to check the impact of the changes or corrections.
system testing	The ability to check the weather and music throughout the entire system.	Script testing based on specifications	This is a test in which a test case is created based on the specifications of the entire system including the interface, and the function is performed manually or automatically. It is recommended that the test cases be divided into two categories: those that clearly show the expected results such as "what time is it" and those that depend on the person or environment such as "play a song that matches the summer". For these evaluation methods, we propose an n-step evaluation in 4.2.6.
		compatibility test	Each company sells multiple variations of products with the same VUI.We will check whether the functions we have developed will work on the respective models of the company.

exploratory testing	In VUIs that deal with natural language, there are multiple ways of saying things to make a particular action happen. This leaves a number of issues that cannot be discovered through specification-based testing. Therefore, it would be better to allow more time for exploratory testing and use theTable 6.2 as a test-charter to confirm these issues.
scenario testing	Each function (weather, open the window, etc.) has its own way of being used. By creating a user-centered story and checking it along the way, there is a possibility of discovering problems.
Field Test	Confirmation at the location where the product will actually be used.Rather than testing the operation of each function, the emphasis will be on confirming the convenience of the system in the location where it will be used.
long-run testing	The music and radio functions are functions that are likely to be used for long periods of time. In the case of car navigation systems, long distance driving is also a possibility. There is also the question of whether a continuous call over a long period of time will result in an error.
Test for accuracy	Although the accuracy at the component test level can be achieved, there may be a problem in VUI where the accuracy is not achieved due to the microphone performance when passed through the microphone of the actual unit. Verify the accuracy using the actual device.
performance test	VUIs, like people, are expected to respond in a timely manner when something is communicated. Each company needs to set its own standards for turnaround time and confirm whether it can respond within those standards.
Security test	There are situations where personal information is handled.In such cases, it is necessary to perform security testing based on expertise. (As with any other product, testing should be performed to confirm the presence of vulnerabilities.
Confirmation test, regression test	When changes or corrections are made, a test to confirm whether the changes or corrections have been made correctly and a regression test to confirm the impact of the changes or corrections on the surrounding area must be conducted.

Pre-release / post-release feedback	The ability to check the weather and music throughout the entire system	user test	Although each function is designed with the user in mind,it is necessary to check how the user will actually use it and where they may be confused.In addition, the way people talk and use the system will change over time. It would be better to conduct user tests with the intended user base and improve the functionality.

Fig. 6.4　Test Architecture (Component Testing)

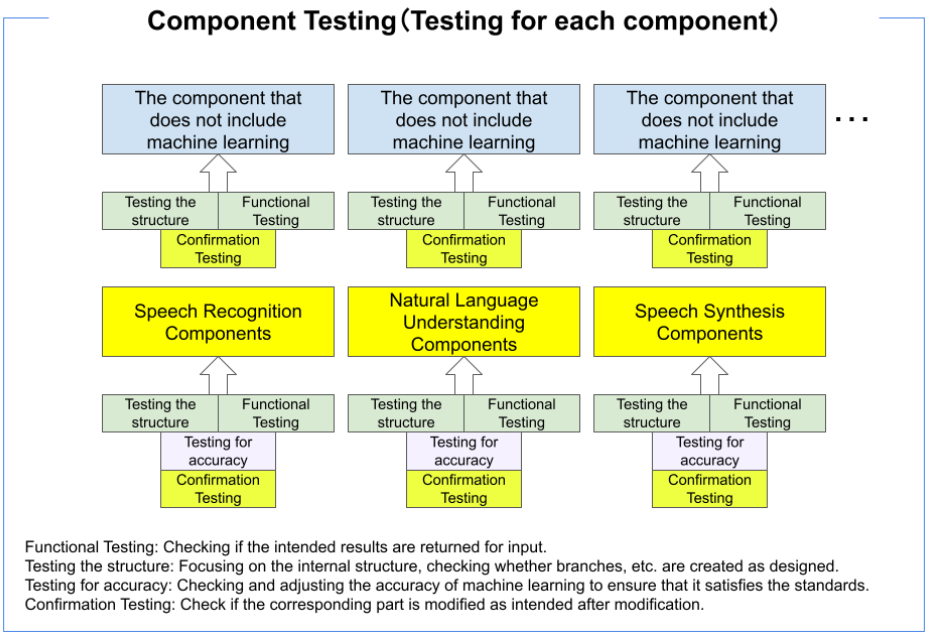


Fig. 6.5　Test Architecture (Integration Testing)

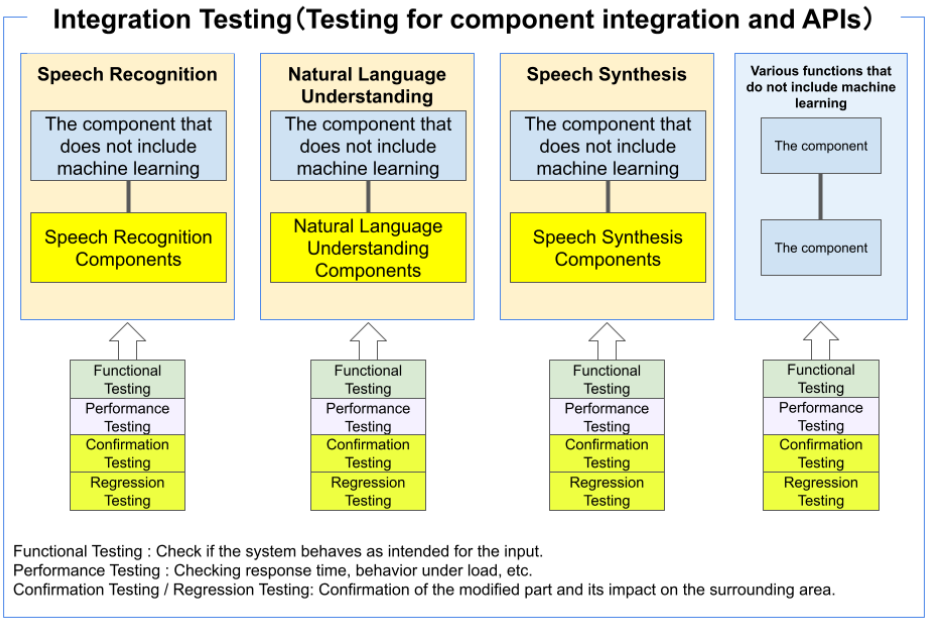


Fig. 6.6　Test Architecture (System Testing)

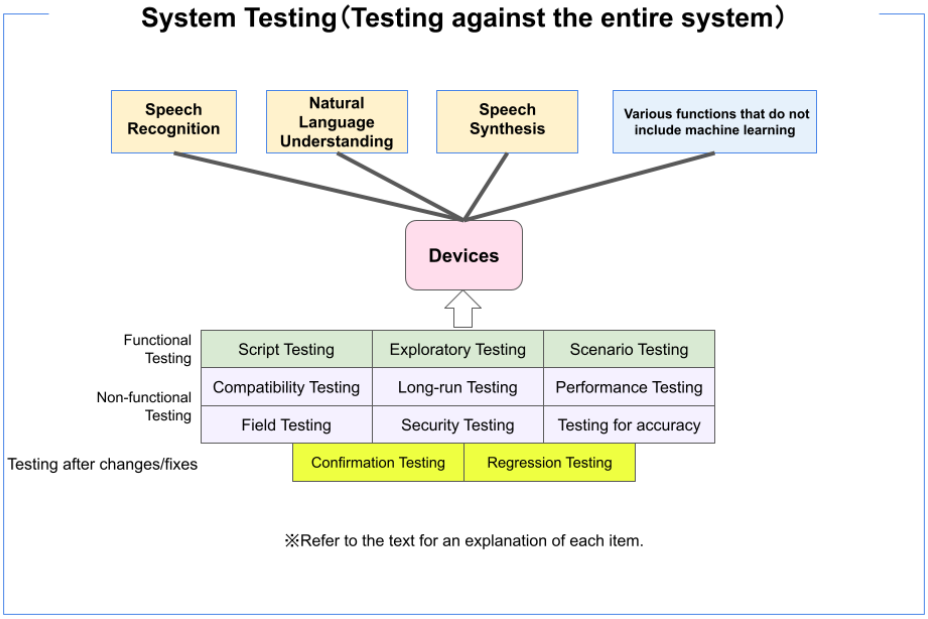


Fig. 6.7　Post-release testing

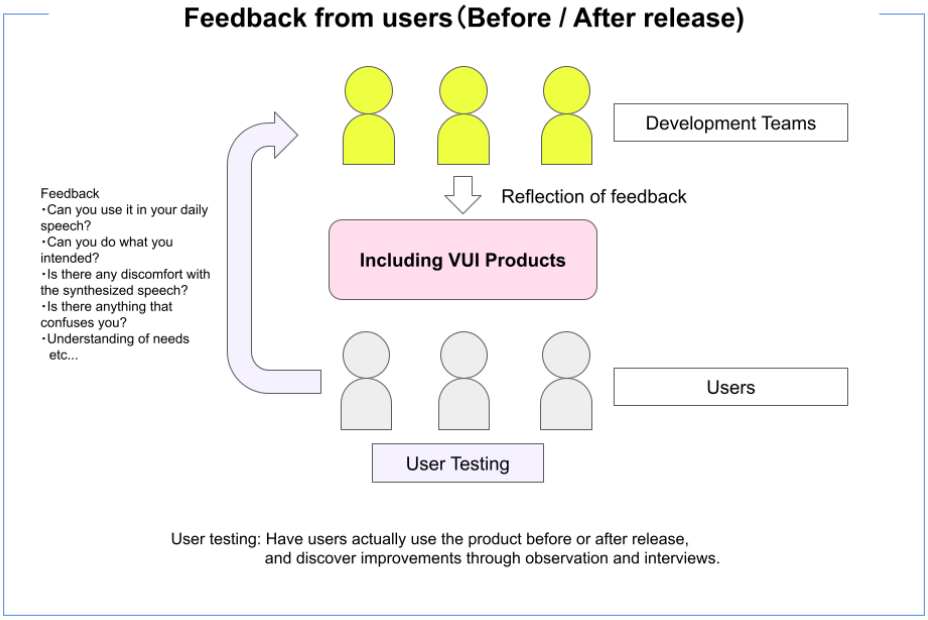


Table 6.2: Examples of patterns needed for learning

The same voice input string can be correctly interpreted as a character string.			
Sound	gender	men and women	-
	Age	High/low	-
	tone	Accent	-
		Rapid speech and breaks	-
		mimicry	Pretentious voices, catcalls, etc.
	feelings	Gentle and strict	Voice with emotion, etc.
	Language beginners	vowel symbol error	The Japanese word "ra" does not distinguish between l and r, etc.
		native language dependency	Turbid sounds, etc. (Korean → Japanese, etc.)
Ability to understand different expressions as standard text.			
text	tone	honorific	-
		command system	-
		young people's language	-
	particle	Variation in "no" and "ha" in Japanese	-
	grammar	Changing the word order	-
		ending a sentence with a noun	-
	acronym	abbreviation	-

	homonym	representation	いどう、はし、あめ等
		ambiguous expression	みたい： looks like/to watch/語尾
		place name	"Shinjuku" and "Ginza" are located all over Japan.
	multilingual	Japanese English	Naive, Tension
	Buzzword	-	-
Guaranteed to work in the operating environment			
Environment	Installation environment	voice noise	TV, neighborhood chatter
		noise	Household noises, opening and closing of doors, wind from fans, etc.
	Installation location	vibration	The scaffolding is bad.
		surface of a wall	Repercussions, etc.
Conveying a voice message that can be understood by the users of the system			
speech synthesis	pronunciation	How to read kanji	"tsuitachi" and "ichinichi", idiom, celebrity
	tone	accent	-
		Rapid speech and breaks	-
		feelings	-
		mimicry	-
	information volume	Importance and urgency	-
Ability to accurately understand the voice input of the system purchaser.			
individualization	speaker recognition	Voice separation	-
		Personal Identification	-

Table 6.3: Examples of perspectives to keep in mind across the test architecture

Examples of perspectives to keep in mind across the test architecture			
Service	Command response time	communication	-
		Interface Compatibility	-
	Service Warranty	Sound Quality	Can be played back as Music data.
	Unauthorized access to services	Parental control	-
	Inappropriate Service Response	indecent	-
		Religion	-
		Taboos in the media	-

		assets	Malfunction around assets and money
		privacy	Malfunction in personal
Calling an external Device			
Appliances Operation	Safety	Safety Affected Device Access	Set the temperature of the bath to 80 ℃,etc.

6.6 Effective methods

6.6.1 N-step evaluation method

When confirming that "speech input is recognized as intended by the speaker and the intended function is executed" as described in the expected quality, it may be difficult to clearly assign a yes/no pass/fail to the expected result depending on the level of abstraction of the question. For example, in response to the question "What time is it? a question with a low level of abstraction, we can expect an output that can be judged by Yes/No. However, it is difficult to judge whether the output is as intended for questions with a high level of abstraction, such as "Play a song that goes with summer" or "Can you tell me a fun story". These questions depend on individual senses.

We propose an n-step evaluation (4-step, 5-step, etc.) as one method to check these. Determine the standard value of pass/fail of the test items of the relevant function, and select the number of survey participants and people. We ask the relevant persons to execute the test items, and ask them to evaluate whether the test items are in accordance with the intentions on an n-step scale. The median or average value of the collected results is taken and compared with the specified standard to verify whether the results are "as intended.

An example of a five-step evaluation is shown below.

1. A different function is executed against the intent
   - : "Fortune telling" is executed for "Play music".
   - "I want to go to a convenience store" will execute "Phone".
2. : The intended function is executed, but a different content is executed within the function than intended.
   - "Stop music" is executed with "forward music".
   - : "Roll down the seat" will cause the seat to rise
3. The intended function is executed, but information/content different from what was intended is returned
   - "Play the song "Singer's Name."" will return "another singer's song".
   - "I want to go to a "destination"" will be set to a "different destination".
4. The intended function is performed and the intended content is returned, but it is not quite right.
   - "Play a song that matches the summer" returns a song, but it is not a standard summer song.
   - "I want to go to my "destination"" returns a different route via a different destination.
5. The intended function is executed and the intended content is returned.
   - "Play a song that matches the summer" will return a "standard summer song".
   - : "I want to go to the "destination"" will return the shortest route.

In this case, 1-3 is the evaluation of the "must-be quality" that should be satisfied as a function. In this

case, 1 to 3 are the evaluation of "must-be quality" that should be satisfied as a function, which is also evaluated as a defect by the user, and 4 and 5 are the evaluation whose result depends on the person and environment.

6.6.2 Smoke test

Speech recognition, natural language understanding, and text-to-speech can be trained to improve the accuracy of targeted functions However, problems occur where the accuracy decreases in other areas. It is necessary to confirm at an early stage whether these problems are occurring in the main use case. For this purpose, we list typical utterances that go through the main use case and verify whether they satisfy the specified accuracy or whether the execution result is equal to the expected result. In the case of this verification, the results can be discriminated as success or failure, and the confirmation is no different from the conventional verification method.

(Example) Smart Speaker

- What's the weather like today?
- The chance of precipitation in the evening is
- Set the alarm for 8:00 a.m. every weekday
- Turn on the radio.

This verification method can be applied to the verification of each module of speech recognition, natural language understanding, and speech synthesis, or to the system test after all modules are installed in the system. In addition, it is necessary to agree on whether to guarantee this verification in each module or in the final system test stage. If this is not done, the number of test man-hours may increase due to duplicate checks, or test omissions may occur due to the belief that the test will be done in one of the processes.

6.6.3 How to evaluate the recognition accuracy of speech recognition

As a method of evaluating the accuracy of speech recognition, we can pick up the elements required for our own project from the elements listed in 6.2.1, combine each of them, and establish a test query and acceptance criteria for verification.

The following is a simplified example for easy understanding. In the example, speech distance, environment, gender, and age are used as elements.

A matrix is created using speaking distance and listening environment. Speaking distance is defined as near, medium, and far (the standard of distance depends on the product). The listening environment is a quiet environment where there is no sound around the VUI system, a noisy environment where there is sound around the VUI system (the standard of dB (decibel) depends on the product), and a vocal environment where the VUI itself emits voice. We ask male and female users in their teens, 30s, and 50s to read out the test query n times in these combinations of speaking distance and listening environment. (The acceptance criteria depend on the product.)

Table 6.4　(Example) Female, 30s: "Today's weather is」

	short distance	middle-distance	long distance
static environment	○	○	○
Noise environment	○	○	×
generating environment	○	×	×

For example, in the case of a car navigation system, the distance is "driver's seat," "front passenger's seat," and "rear passenger's seat," and the environment is "quiet environment," "environment while driving," and "environment with the window open while driving," etc. Note that the contents to be considered vary depending on the product.

6.6.4 Natural Language Understanding Test Case

There are as many input values to the natural language understanding module as there are ways to say them. The input words are very diverse, but they are intended to be the commands we want to execute. Therefore, when we perform black box testing of the natural language understanding module, we can prepare the expected results along with determining the input values to be tested.

When creating a test case, we can pick up the elements that are necessary for our project from the elements listed in the natural language understanding section of 9.3.4, and combine them with keywords that are important for the corresponding function (for example, weather and date for the weather function). The combination may be done by using keywords that are important for the function (for example, weather and date for the weather function). In consideration of the risks involved, it is necessary to consider where the combination should cover only one factor and where a combination of two or more factors is required.

In addition, when testing the natural language understanding module, the number of test cases is huge. It is difficult to perform this testing manually, and it is recommended to automate the testing of the module. This test is to run the necessary test cases for initial accuracy testing as well as regression testing for future updates.

As an example, table6.6 describes how to create a test case for the weather notification function for a specific date and time.

6.6.5 Internal user testing

In AI products, there are many areas where the results cannot be determined uniformly depending on the test conditions. Therefore, the user testing method is not only effective for improving UX quality, but also for AI quality. Especially when the AI model is Deep Learning, there is a high possibility of encountering cases that are not expected in the training data.

For the above reasons, it is necessary to conduct tests under the conditions that match the usage environment envisioned for the product. (For example, testing in a quiet environment in the company when the product is intended for daily use does not meet the purpose of the test.

In addition, it is assumed that assigning personnel who are not normally involved in testing to conduct this test will be more effective. However, since these personnel have little experience in testing, it is necessary to fill in the gaps in their understanding of the purpose of the test, such as what kind of test

Table 6.6 Example of how to create test cases for natural language understanding functions

Creation method	Description.	Example
(Basic)	Prepare the basic test case which contains the keyword elements decided by the specification.	What's the weather? What's the weather like tomorrow? What's the weather like in Aomori tomorrow? What will the weather be like in Aomori tomorrow at 3:00 p.m.? (day + time + place + question)
word order change	Change the word order of the test case and create a new test case. The expected result will be the same as the basic test case because Japanese has a metamorphic relationship where the meaning does not change even if the word order changes.	・What is the weather like tomorrow in Aomori City? (place+day+question) ・What is the weather like tomorrow at 3:00 PM in Aomori City? (day+place+time+question)
word change	Create a new test case by changing the keyword words in the test case. When selecting words, we consider words that are important or have a high risk of failure depending on the test target. For example, in the case of the weather function, we would select a location with a large population.	What's the weather? → Will it be sunny? /Rain? /Snow? etc. Tomorrow → day before yesterday/today/tomorrow/after tomorrow/April 1st/3 days later etc. Aomori City → Yokohama City/Osaka City/Nagoya City/Sapporo City etc. 3pm → 3pm/0am/22pm etc.
Changing the end of a word	Change the ending of the test case and create a new test case. If the product has already been released, analyze the logs to see what kind of endings are common. If it has not yet been released, create a persona of the target audience and envision how they will speak.	What's the weather like tomorrow? What's the weathow?
particle modification	Change or delete a particle in a test case and create a new test case. In spoken language, it is common for particles to be incorrect or missing.	・What's the weather like tomorrow? What is the weather like in Aomori City tomorrow?

and what aspects should be checked.

The purpose of the test will not be achieved if the reports of the test participants are biased towards "it works fine". Therefore, it is recommended to inform the test participants of the following points in advance.

- How accurate is the speech recognition?
- Are the responses conversational and natural?
- Has the introduction of the AI product impacted your life?
- Has it created a new experience?

6.6.6 Evaluation of accuracy when data is changed, or when model is changed

When data is changed or model is changed, how the accuracy is changed by the change is evaluated. If both the data and the model are changed during this evaluation, it will be impossible to isolate which change caused the change in accuracy.

Therefore, if you want to check the accuracy by changing the data, fix the model and measure the accuracy by passing the data before the change through the corresponding model. Then, the data after the change is passed through the same model to measure the accuracy. By doing this, you can evaluate what kind of data is more appropriate for the model.

Similarly, if you want to check the accuracy by changing the model, fix the data. By passing the data through the model before the change and measuring the accuracy, and then passing the same data as before the change through the model after the change, we can evaluate which model has better accuracy for the corresponding data.

6.7 Quality Assurance Level

The following two levels of system-wide quality assurance for VUI systems are possible. The details are shown in figure 6.8.

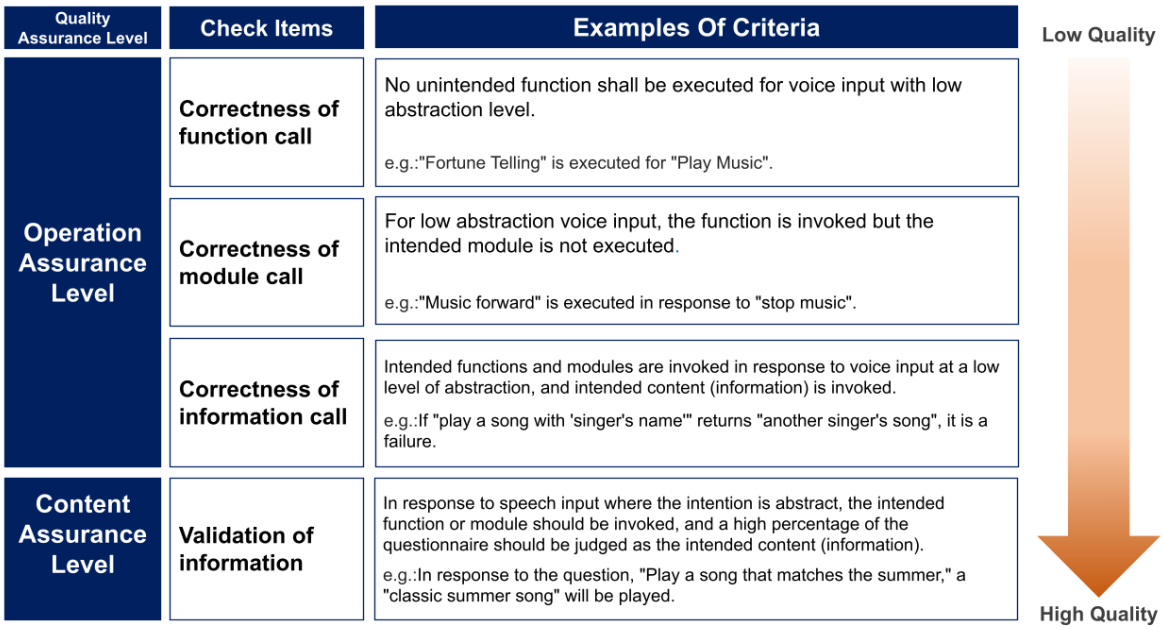
1. **Operation Assurance Level**
   The result of the test of the must-be quality part that can be answered with Yes/No (in the example of the five-level evaluation in 6.6.1, the area of 1~3) meets the specified acceptance criteria.
2. **Content Assurance Level**
   The results of the attractiveness quality portion of the test (i.e., the area of 4~5 in the example of the five-step evaluation of 6.6.1) meet the specified acceptance criteria.

In the aforementioned five-level evaluation example, the area of 1~3 is defined as the operation collateral level, and it is classified as a level of quality assurance only when all of them are collateralized. The quality secured at this level ensures that the functions, modules, and information called from the input speech information are correct, respectively. It is assumed that the input voice information can be judged by Yes/No, and that the intended result is returned from the assumed information.

On the other hand, the content assurance level defines a domain of 4 to 5 in the aforementioned five-step evaluation example. Therefore, the assumed audio information is at a high level of abstraction, and a questionnaire is used to evaluate whether the returned results are intended or not. It is recommended that the results of the questionnaire be evaluated at the content assurance level, and that acceptance criteria be established and evaluated for each development organization or project.

.

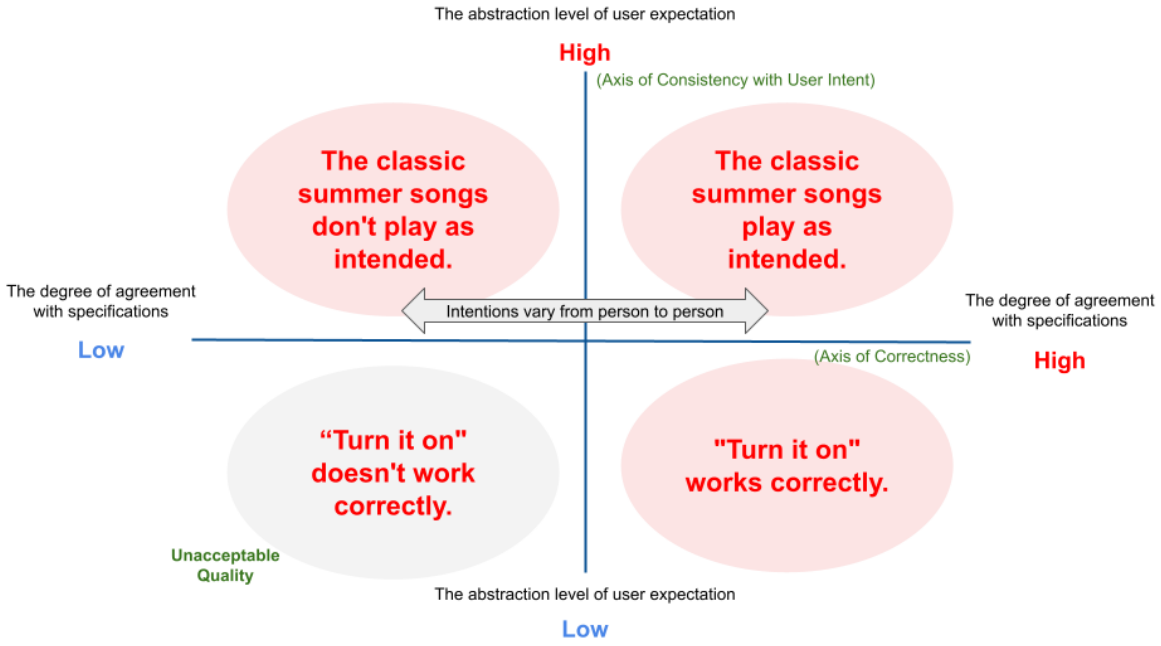
Fig. 6.8　Quality Assurance Level for the VUI as a whole



When applying these quality levels, it is important to isolate the level of abstraction.

As a guideline for isolating the level of abstraction, it is recommended that the level of conformance with the specification (what can be defined as a specification) and the level of abstraction of human expectations (whether or not the expected result varies from person to person) be used as criteria. An example is shown in Figure 6.9. By separating the expected results of the voice data to be used in the test as shown in the figure below and separating whether the test to be conducted is at the operation assurance level or the content assurance level, it will be possible to evaluate the quality level appropriately.

Fig. 6.9　Examples of guidelines for isolating abstractions



In addition, in the case of machine learning, in order to guarantee the quality of the data, it is important not only to check the output data but also to guarantee the quality of the teacher data used as the training source.

The quality assurance level of the teacher data can be divided into two categories

- Use of unspecified data (especially uncategorised data)
- Data identified in each element described in 6.2.

For each of the elements of speech recognition, natural language understanding, and text-to-speech functions listed in 6.2, we believe that by setting a range and criteria for the extent to which each organization should guarantee these functions, it will be possible to classify the level of achievement in more detail. We believe that this will enable a more detailed classification of the level of achievement.

7. Industrial Processes

7.1 Assumptions and targets for consideration

In this chapter, we will focus on quality assurance of AI in industrial systems, using control systems as an example. In industrial systems, control technologies such as statistical quality control and feedback control have been developed for the purpose of stabilizing quality and improving productivity. In recent years, machine learning technology has been applied and put to practical use in many fields, such as automatic identification represented by images, and the introduction of anomaly detection and change-point detection for the purpose of predictive maintenance of equipment. Figure 7.1 shows an example of an industrial system application and the functions that apply AI technology.

In examining quality assurance in such a variety of industrial systems, plant control was the main subject, and the study was conducted in the following order.

First, the 7.2 section clarifies the key issues to be considered in applying AI technology to industrial systems. Next, in 7.3 section, the basic architecture of an industrial system incorporating AI technology

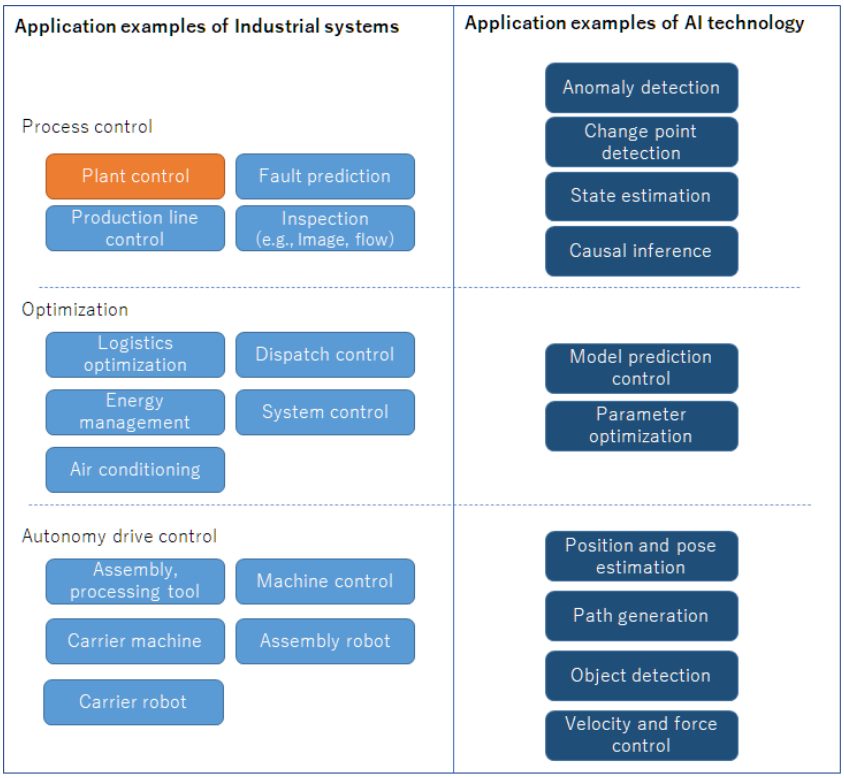


Fig. 7.1 Examples of industrial system applications and application of AI technology

is presented, and for each of its components, the considerations for the key issues identified in 7.2 section are presented. The 7.4 section indicates the assumed stakeholders. In addition, the 7.5 section shows the process and detail process from PoC to operation of the industrial system applying AI technology, and for each process, the considerations for the key issues shown in the 7.2 section are shown. Based on the discussion in the 7.2-7.5 sections, the 7.6 section will specify the quality assurance considerations for industrial systems based on five indicators (Data Integrity, Model Robustness, System Quality, Process Agility, and Customer Expectation). The correspondence between the process and the five indicators in the industrial system embodied in 7.6 section is shown in 7.7 section, and an example of a quality assurance study based on a published case study with a concrete image of the application of AI technology is shown in 7.8. Based on the above, the flow of a quality assurance study for an industrial system applying AI technology. The flow of the quality assurance study for industrial systems applying AI technology is shown in Figure 7.2.

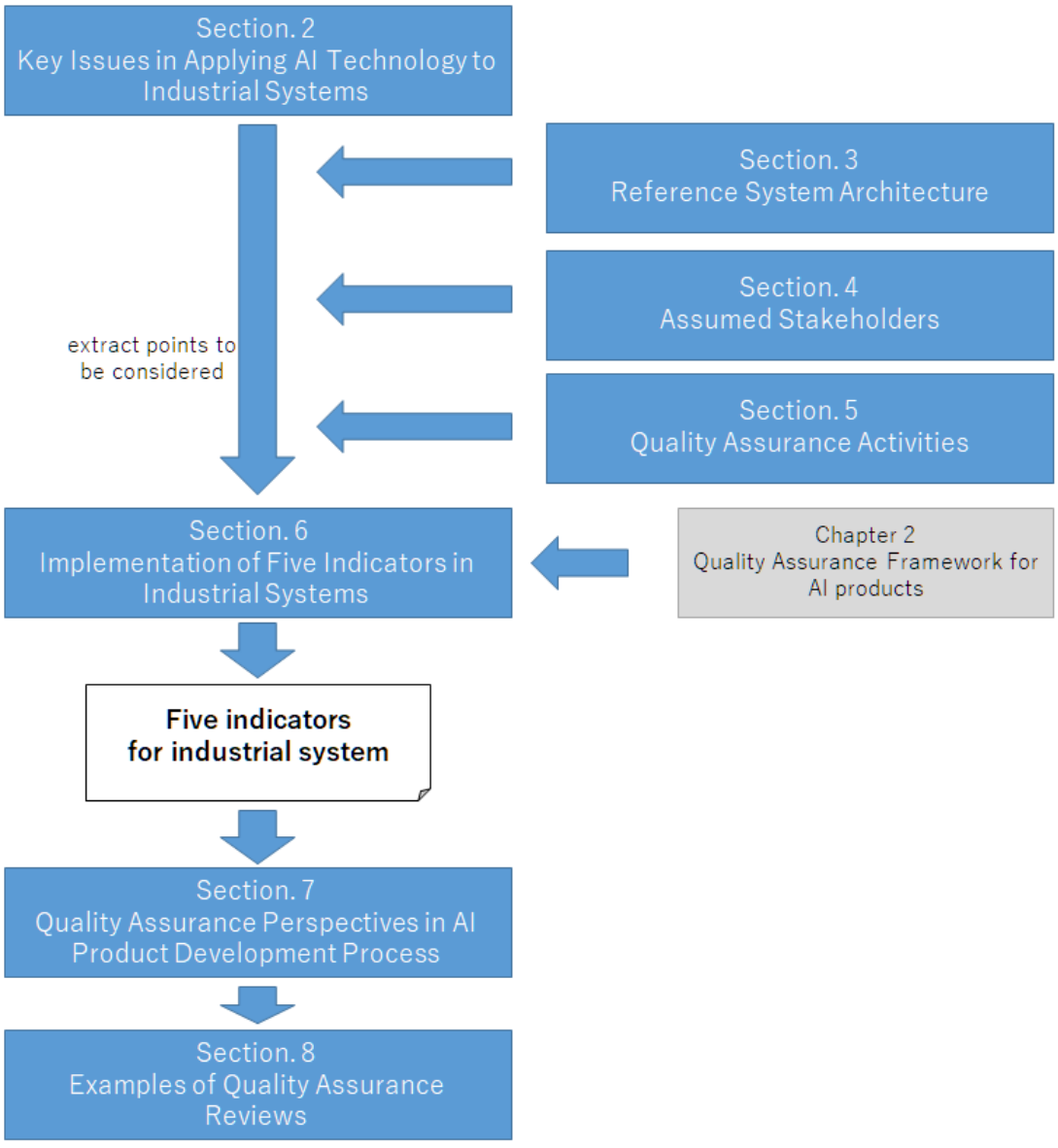


Fig. 7.2　The flow of the industrial system quality assurance study in this guideline

7.1.1 definitions of terms used in this chapter

In this chapter, each term is used with the meaning shown below. The definitions of the terms are based on the Contract Guidelines for the Use of AI and Data [4].

7.2 Key issues in Applying AI technology to industrial Systems

In industrial systems, quality assurance requires not only the characteristics of the AI technology itself, but also the quality objectives of the target system, such as "reliability" and "safety," as well as problem solving based on the characteristics and constraints of the system.

Based on this, the following three priority issues for quality assurance of industrial systems can be identified.

1. Stakeholder diversity: Because of the large scale and complexity of the system, there are many forms of subsystems that are built and operated by multiple providers based on contracts. It is necessary to verify the integrity of individual data, rights protection, and the system as a whole.
2. Environmental dependency: Systems are exposed to 5M+E changes and need to be guaranteed based on the assumption of diverse data and data with different reproducibility.
3. Explainability: There are multiple processes and standards that assure the validity of the system and need to elicit accountability and buy-in from the customer.

Therefore, the Industrial Process WG developed the guideline to improve consensus on the issues by organizing the viewpoints in the balance chart based on the five indicators and the process (PoC/Development/Operation) of AI products based on the QA4AI Guide.

The Industrial Process WG expects the guideline users to expand based on the requirements and characteristics of the target AI products based on this study, and to promote consensus building in planning and implementation with related parties.

* 5M+E is an abbreviation for Man, Machine, Method, Material, Measurement and Environment. It is a term used in quality control in machining and factories. 5M+E refers to the typical control items for quality change factors, such as man, machine, method, material, inspection/measurement, and environment.

7.3 Reference System Architecture

The Industrial Process WG has prepared a reference system that abstracts the system architecture in order to typify the quality assurance activities of various systems (Figure 7.3).

Reference System Architecture

Real-world industrial systems are composed of this reference system, which is connected and hierarchical.P1 to P3 are subsystems of the existing system to which the AI component is to be added. P1 is the input part for the AI component, and P3 is the output part. P2 is the control processing part, which is an alternative for the AI component.

The following extensions to the probabilistic behavior of the AI component are required.

- P1: Assurance of data assumptions, accuracy, and label validity required by the AI component
  addressing stakeholder diversity　Ensuring data integrity
  addressing environmental dependency　Assurance of data interval and range

Table 7.1 terminology definition

Terms	Definition
AI	The term "machine learning" refers to the process of making machines do what humans do with their intelligence, especially when it is achieved using "machine learning.
Machine Learning	Machine learning is a general term for a method that discovers certain rules in data and mechanically performs inferences and predictions for unknown data based on those rules.
AI systems	systems that use AI to achieve their goals. It is synonymous with AI product. In the example in 7.8section, the packaging machine is the AI system.
AI Products	Synonymous with AI systems.
AI Components	A component of an AI system that uses AI to perform some processing. In the example in 7.8section, it is the film snaking detection part.
AI Algorithms	A method or procedure for performing machine learning. A typical AI algorithm is a neural network; an AI model is generated by applying training data to an AI algorithm.
AI models	The part of the AI component that performs inference and prediction from input data. It is composed of neural networks and weight parameters. It is generated by machine learning, and the results are reflected in the weight parameters.
raw data	Data that has been obtained from sensors, providers, etc., and has been converted and processed so that it can be read into a database. (In many cases, the data is not suitable for learning as it is because it contains missing values or outliers.
Data set for learning	A collection of raw data that has undergone secondary processing, such as preprocessing to remove missing values and outliers, and labeling. It consists of training data and test data.
Training data	Data included in a training dataset that is used to perform machine learning on an AI model.
test data	Data included in the training data set that is used to check the generalization performance, etc., of the trained AI model.
Operational data	Data to be input into the AI model during actual operation
input data	Data to be input to AI model for inference and discrimination
Output data	Output when input data is given to AI model.
learning program	A program that performs machine learning.
AI Program	A generic term for the learning program and the program used in the AI component.
Hyperparameters	A parameter used to define the framework of machine learning, such as the learning rate, the number of learning epochs, or the Drop Out parameter, which is mainly determined artificially.
Re-learning	After the AI model is generated, the training program itself is left unchanged, but the training data set and hyperparameters are changed again, and the model is retrained from the beginning.
Additional learning	Applying different training datasets to an existing AI model for further machine learning.
distillation	Generate new AI models that are smaller and simpler by using the input and output data of existing AI models as training data sets.

Explainability　Data recording to explain the behavior of AI components

Using image inspection as an example, P1 is responsible for assuring proper brightness, illumination angle, focus, etc., and storing the target image data as needed.

In plant control, it is responsible for ensuring that the sensor values of the target material are valid.

- P2: Alternative systems and monitoring of AI components
  Dealing with environmental dependencies　Monitoring and observation of events that may not be subject to the operational assumptions of the AI component
  Explainability　Comparing and Ensuring Operational Performance of AI Components

  P2 is an example of image inspection, as shown below.
  Example 1: A person is responsible for sample inspection, which is responsible for evaluation against environmental dependencies, and monitoring of the AI component.
  Example 2: Monitoring by different rule-based metrics other than image inspection (e.g., location, timing, inspection equipment, etc.).

  In the case of abnormality detection in plant control, P2 monitors sensors and control data for abnormalities based on rules, while the AI component uses machine learning to monitor abnormalities. The AI component estimates that an abnormality has occurred and outputs the result.

- P3:Guaranteed output for the results and accuracy of the data output by the AI component.
  Addressing stakeholder diversity　Clarify how AI component output values contribute to P3 output.
  Environmental dependency　Filtering and judging AI component output based on P2 information for environmental dependency

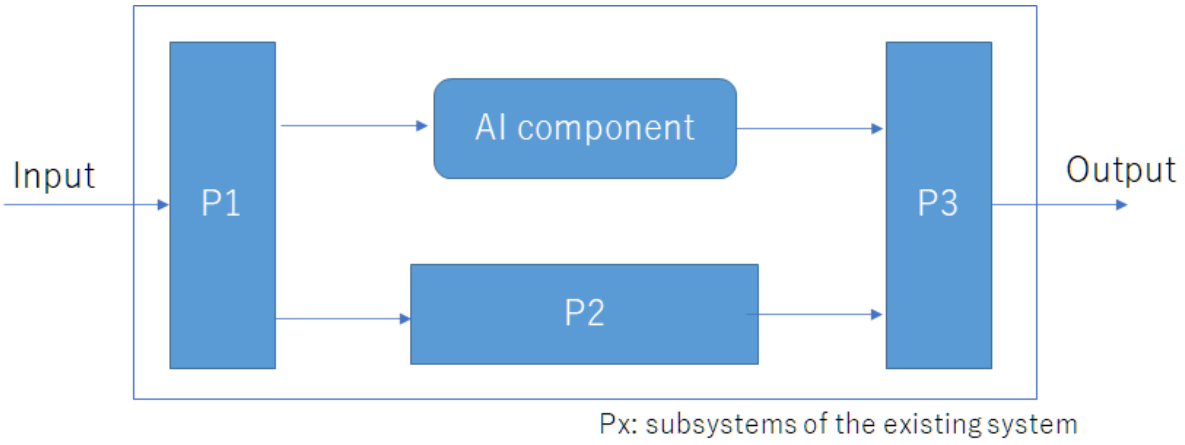


Fig. 7.3　Reference System Architecture

For ease of explanation　Record AI component output and decision results

Using image inspection as an example, the AI component outputs results such as "x % probability of belonging to the good set" and "y % probability of belonging to the bad set". Based on the rules, the AI component determines how to use these results for the entire system. (For example, prepare threshold values for x and y.)

Using plant control as an example, there may be a case where P3 outputs correction commands to move the system to the optimal state based on the system state recognized by the AI component. In order to demonstrate to the various stakeholders that this is a valid correction order, the relationship between the state recognized by the AI component, environmental dependencies, and system quality must be explained based on some sort of empirical means.

At the time of writing this guideline, there is no way to guarantee the assurance of systems that use machine learning technology in the architecture for these issues. It is necessary to solve these problems for each individual system based on the points to be considered in quality assurance. Therefore, this WG has organized the points to be considered in the five axes of quality assurance activities for the guideline users to use in their quality assurance activities in actual systems.

7.4 Assumed Stakeholders

In the Industrial Process WG, stakeholders are defined as those involved in AI product development. Stakeholders are divided into two groups: manufacturers who develop AI systems and customers who install AI. The customer side is mainly factories, which are expected to apply AI to improve their productivity. The correlation of the stakeholders is shown in Figure 7.4 and their roles are described in Table 7.2.

In the above, we have assumed that the system operation is performed by the customer. It is also possible that the manufacturer will operate the system and the customer will only receive the service. In this case, the system operator is on the manufacturer side. On the other hand, on the customer side, there is a service user section, which includes the service user section manager who has the requirements and the service users who actually use the service.

7.5 Quality Assurance Activities

In industrial systems, in order to satisfy the required quality, the system development process is broken down into processes, and quality is guaranteed by systematically verifying that each process is appropriate (process control). We believe that it is useful for quality assurance to achieve quality assurance of AI products in relation to the five indicators for each process, based on the concept of process quality in actual systems.

This WG defines the entire process of industrial systems as the IXI (Intelligent eXperimental Integration) model, which defines the entire development process, referring to the process form of SQuBOK®[2]. It is defined as shown in reffig:IXImodel. This process is broadly divided into three processes: PoC, development (including the development process of the machine learning program),

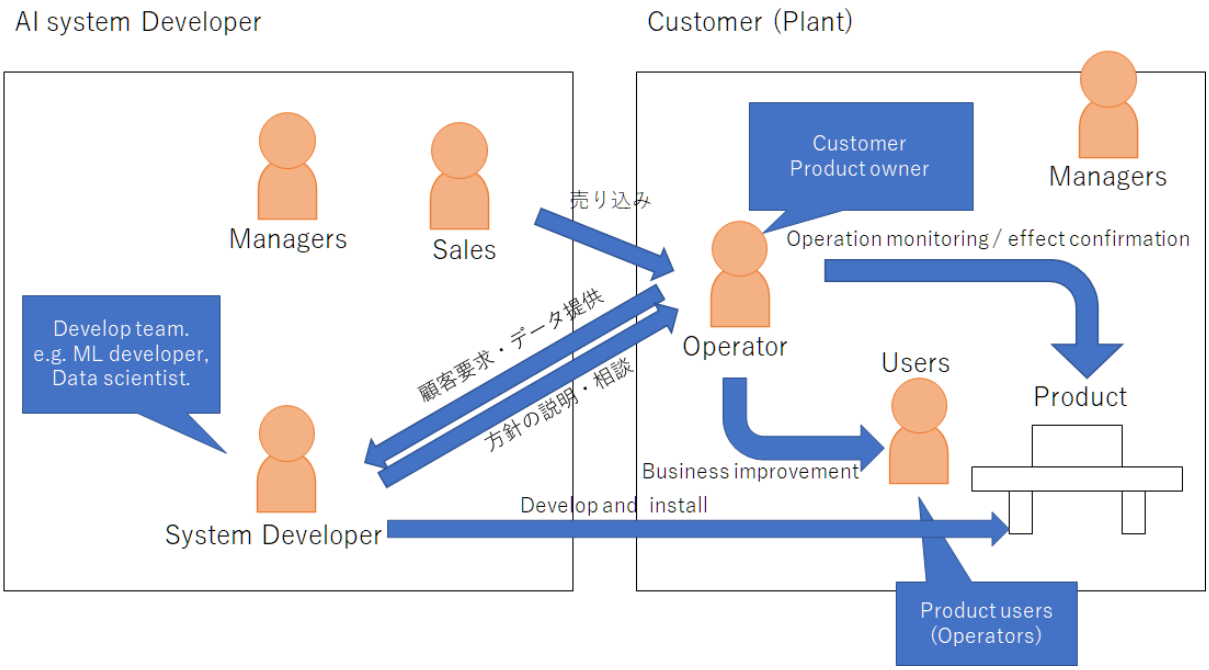


Fig. 7.4　Stakeholder correlation chart

and operation. "In PoC, the main risks in development and operation are identified and verified. In the "Operation" phase, we will monitor the output of the developed system, collect data necessary for explaining the output results to stakeholders, and evaluate the system using the data confirmed for the first time during operation, and update the model as necessary. The light-orange background box in Figure 7.5 indicates activities that are particularly important in the development of AI-based products.

Table 7.2　Role of Stakeholders

Stakeholder Name	Key Roles
Managers (AI system developer)	managing the development team.
Sales	Marketing Planning and explanation of productivity improvement using AI systems to customers, and promoting the introduction of AI systems.
System Developers	Design and develop AI systems. In the figure, it is only arranged as a system developer, but in reality, it often consists of many members such as architects, data scientists, and other specialists, as well as subcontractors who are in charge of implementation, and it is necessary to consider the stakeholder diversity described in the previous section.
Managers (Customer)	AI system with authority and responsibility for implementation. He is the manager of the system operator.
Operator	In charge of factory line management, AI system installation and operation. They have their own requirements for the implementation of the system, and their job is to improve productivity. From the manufacturer's point of view, they are customers.
Users	are the people who will actually use the deployed AI system.

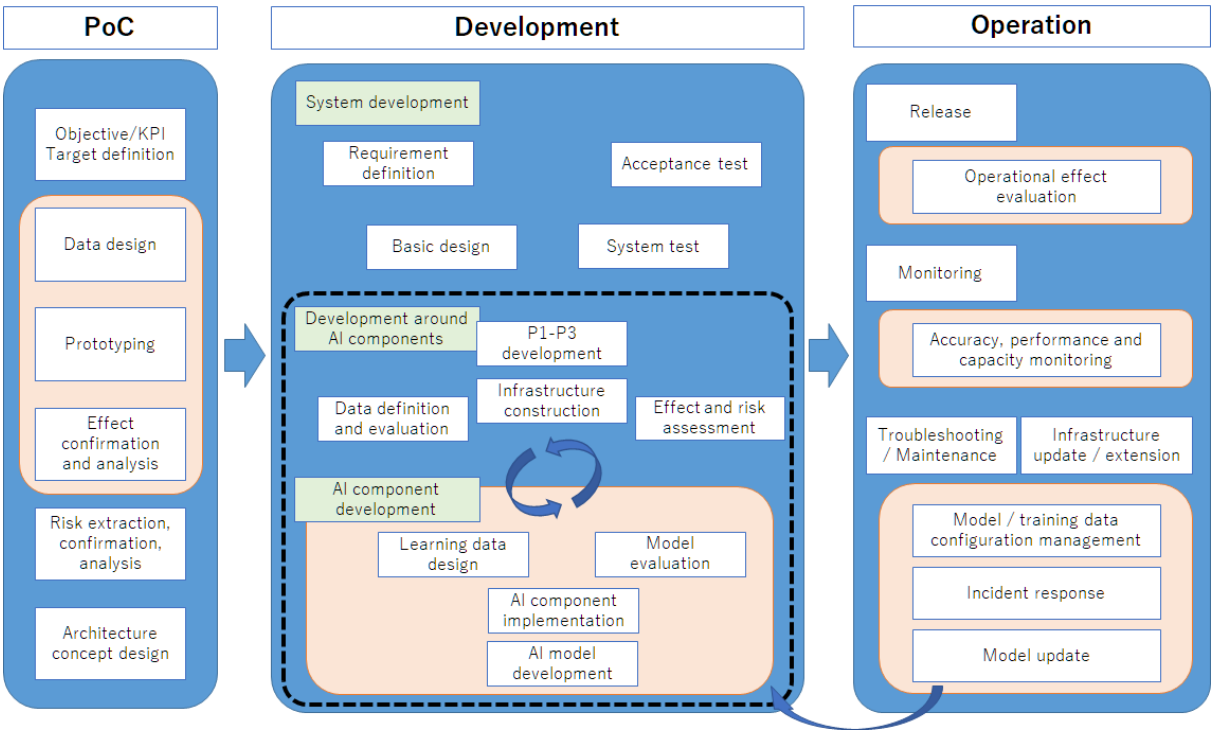


Fig. 7.5　Overview of the process：IXI: Intelligent eXperimental Integration model

Table 7.3 Organizing key issues through a three-step process

Engineering	PoC	Development (Dev.)	Operation (Ops.)
Objective	Identify achievability and feasibility and reach development agreements with diverse stakeholders	Based on PoC results and results under development, establish assurance items and methods for environmental dependency, and reach operational agreement	Operate the AI system in the field, evaluate and respond to performance and occurrences, and stabilize the operation
Stakeholder Diversity	**Agree on goals, risks, etc. with stakeholders and demonstrate with prototypes**	Interface/API design and data consistency support	Support for on-site operational requirements, change management requirements, etc.
Environmental dependency	some environmental conditions based targets and risk assessment	**Clarification of environmental conditions to be covered by the system, and development and verification of the system**	Monitoring and data collection/evaluation of out-of-spec environment
Explainability	Identifying explanatory requirements and reflecting them in system requirements	Explanation of model evaluation and code quality with configuration-controlled environment-dependent data	**Explanation of reasons and countermeasures based on data, model, and configuration when events occur and when performance changes**

By organizing these three processes into the three key issues shown in the 7.2 section, the considerations for industrial systems can be extracted as shown in Table 7.3. This classification shows that the key issues are stakeholder diversity for PoC, environmental dependency for development, and explainability for operation.

When the IXI model is applied in practice, for example, in PoC, Customer Expectation and Process Agility among the five indices will be higher if there is more emphasis on dealing with stakeholder diversity. In addition, during development, except for events that can only be discovered during operation, system development is carried out while considering environmental dependencies to increase Data Integrity and Model Robustness, and to raise System Quality to a level equivalent to production. As the system is put into operation, data and models are acquired in the field and adapted to the field operation, the balance chart of quality assurance by all five indicators is prepared, and the degree of quality assurance of the industrial system applying the provided AI technology can be clearly indicated.

An example of the changes in the balance chart of quality assurance is shown in Figure 7.6.

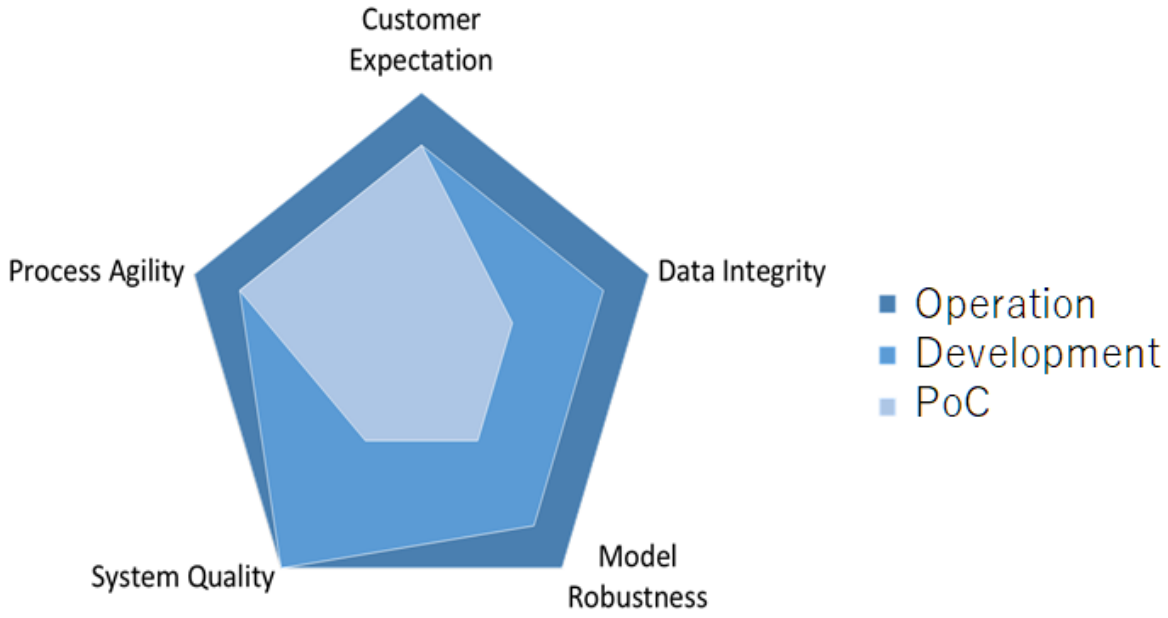


Fig. 7.6　Example balance chart of quality assurance levels

Specific level measurement methods and scales need to be concretized with reference to this guideline. Please refer to the association of individual activities in the development process (7.7 section) and examples of studies (7.8 section).

7.6 Implementation of Five Indicators in Industrial Systems

7.6.1 Interpreting the QA4AI's Quality Assurance Perspective

The weighting of the interpretation of the five axes specified in QA4AI differs for industrial systems because the persons responsible for implementation and the persons involved in the scenes of PoC (goal setting, prototyping, and effect verification/analysis), development (design, manufacturing, and installation), and operation (operation and maintenance) are different. Therefore, this WG will present the points to be considered (interpretation method) for the 5 axes of quality assurance activities and clearly specify the target scenes (PoC, development, and operation) for each point to be considered.

In addition, each point will be explained in a concise manner so that the reader can recall specific quality assurance activities, and the impact of neglecting the points will also be described.

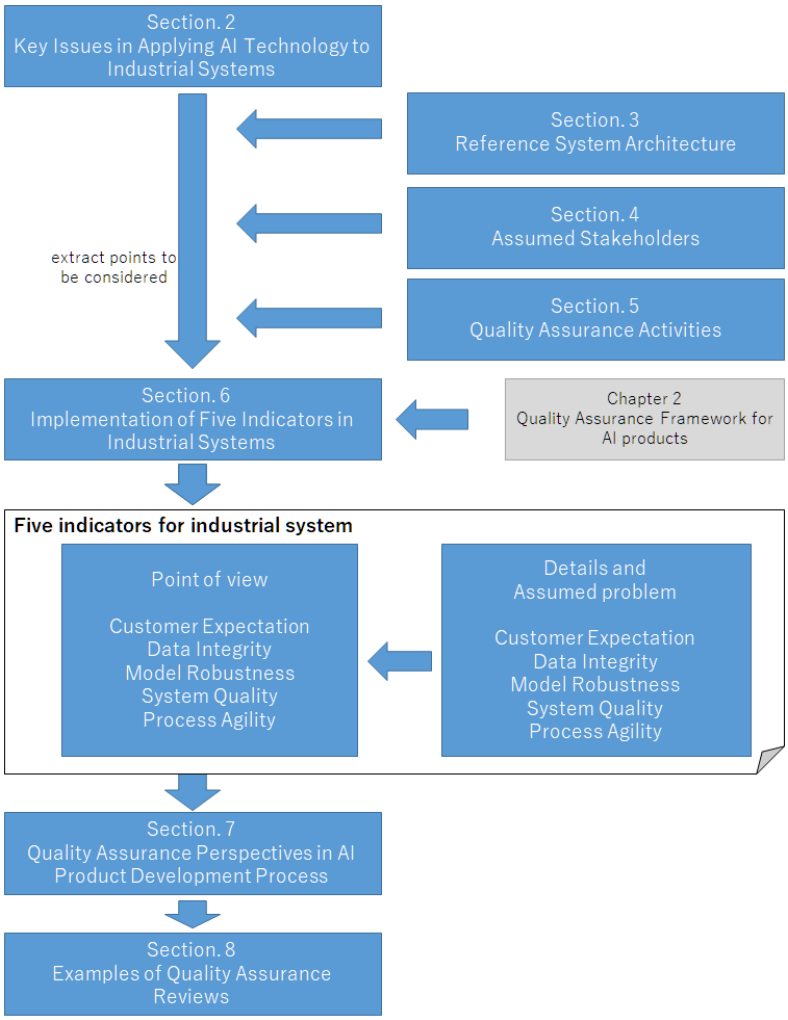


Fig. 7.7　5 indicators embodied in industrial systems

7.6.2 Customer Expectation

Industrial systems have complex interests due to stakeholder diversity. As a result, it is necessary to adjust not only the conflicting expectations of AI systems, but also the rights and commonality of intangible assets necessary for machine learning, such as data models.

Table 7.4 Considerations for Customer Expectation

ID	Point of view	Considerations	PoC	Dev.	Ops.
CE-1	High customer-side expectations Whether it's "human-like" that you're aiming for or not.	● Clarify the business problem to be solved by AI ・Is the customer's business problem clear? (Isn't the application of AI the goal?)	○	○	
		● Possibility of solving business problems with AI ・Can AI solve the business problems of customers?	○	○	
		● Effectiveness of solving business problems to be solved by AI ・Are the customer's expected effects (target performance, etc.) of using AI clear? ・Do you expect them to be as effective or better than "humans?"	○	○	
		● Level of understanding of continuous improvement in maintaining the performance of AI ・Do customers understand the need for continuous learning and improvement in order to maintain AI performance?	○		○
		● Satisfaction with solving business problems through AI ・Are customers satisfied with the results of AI-based problem solving?			○
CE-2	Non-acceptance of the idea of stochastic behavior	● Understanding of AI output with probabilistic behavior ・Does the customer understand that the output results of AI are produced by probabilistic behavior (probabilistically plausible solutions)?	○	○	○
	Inadequate countermeasures due to lack of understanding of risks and side effects on the part of customers and easy demand	● Risk tolerance of the results output by the stochastic behavior. ・Is it possible to allow the results output by stochastic behavior (probabilistically plausible solutions)?	○	○	○

Table 7.5　Considerations for Customer Expectation (cont'd)

ID	Point of view	Considerations	PoC	Dev.	Ops.
CE-3	Closeness The degree to which they do not understand the concept of PoC or beta release	● Level of understanding of agile software development ・Does the customer understand that the development will be done in an inductive agile style, which is different from the traditional deductive waterfall style development?	○	○	
		● Anderstanding of continuous improvement for maintaining the performance of AI systems ・Do customers understand that AI systems need to continuously learn and improve in order to maintain their performance?			○
		● Level of understanding of verification methods for post-operation improvements ・Does the customer understand that post-operation improvements will not be made by modifying the program, but by additional training/re-training of the training data set, adjustment of hyperparameters, etc.?			○
CE-4	Lack of awareness of the quantity and quality of data	● Understanding of the need for a training dataset that matches the customer's business problem ・Does the customer understand that training datasets that match the customer's business challenges are needed to improve the performance of the AI model? Do they have such training data sets?	○	○	
		● Level of understanding of the quality and quantity of training data sets required for training AI models. ・Do customers understand that solving business problems with AI systems requires the exhaustive preparation of many training data sets?	○	○	
		● Customer understanding of changes in input data trends during operation ・Do you understand and agree with the customer that the existing AI model may not be able to infer correctly if the input data trend becomes different during operation from that during training? Also, do you constantly monitor input data trends during operation and share the information with the customer?	○	○	○

Table 7.6　Considerations for Customer Expectation (cont'd)

ID	Point of view	Considerations	PoC	Dev.	Ops.
CE-5	Whether there are any legal or ethical issues in the use of AI products、	● Contractual arrangements regarding intellectual property rights, including copyrights for AI systems. ・Is there a contractual agreement with the customer regarding the ownership of intellectual property rights, including copyrights, and conditions of use for the AI system?	○	○	
	Necessity of consideration for the privacy of third parties, and degree of social acceptance of the use of AI products	● AI Level of understanding of intellectual property rights, including system copyrights ・AI Does the customer understand the ownership of intellectual property rights, including copyrights, and the conditions of use of the system?	○	○	
		● Level of security of customer data used in AI systems, scope of information disclosure and clarification of handling restrictions ・Is the level of security, scope of information disclosure, and handling restrictions of the data used in the AI system clear? ・Is the level of security, scope of information disclosure, and handling restrictions of the data used in the AI system clear?	○	○	
		● Arrangements regarding rights to data contained in AI systems Are data rights and usage rules clear for all data directly or indirectly related to the AI system, including training data sets, test data, input data during operation, and output data?	○	○	
		● Contractual arrangements regarding intellectual property rights, including copyrights for AI systems that have been improved after operation. ・Do changes to the AI model (e.g., additional learning/re-training) made after operation conflict with the rights ownership and usage conditions in the contract at the time of introduction?(For example, if the learning model includes intellectual property rights owned by the developer/provider, is consideration given to updating the model by the user?) ・Is there a contractual agreement on the ownership of intellectual property rights and conditions of use for AI systems that have been improved after operation through re-learning, etc.? ・Are the rights ownership and conditions of use clear for data from customers and users that are input during operation?			○

Table 7.7　Considerations for Customer Expectation (cont'd)

ID	Point of view	Considerations	PoC	Dev.	Ops.
CE-6	Degree to which "rational" explanations are sought, and degree to which "extrapolation" and "prediction" are sought "Cause The degree to which customers are willing to seek "cause" or "responsibility" A climate or atmosphere that empathizes with a sense of conviction, and a lack of work procedures	● The degree to which customers understand the difficulty of explaining the results of AI models Do customers understand the difficulty in explaining the results of AI models?	○	○	○
		● Understanding of explanatory difficulty and accuracy of AI models ・Different types of AI algorithms can provide explanatory evidence for output results (e.g., binary trees), while others have difficulty (e.g., DeepLearning), and whether customers understand that explainability can be inversely proportional to the accuracy of AI models. Does the customer understand that explainability may be inversely proportional to the accuracy of the AI model?	○	○	○
		● Level of understanding of the choice of AI-model Do you agree with the customer that the AI algorithm and model selected during [PoC] and [development] is the right choice? Do you explain the rationale behind your choices and is the customer able to understand them?	○	○	
CE-7	Is the responsibility clear (contract stating that the developer is responsible for the accident)	● Clarification of customers and stakeholders Do you identify the stakeholders required for the AI system [during PoC], [during development], and [during operation], and clarify how each is involved in the AI system? and how they will be involved in the AI system?	○	○	○
		● Clarification of responsibility for the output of AI systems ・When human suffering occurs as a result of the output of an AI system with probabilistic behavior (probabilistically plausible solution), is the responsibility of the system clarified during [PoC], [development], and [operation]?	○	○	○

Table 7.8　Considerations for Customer Expectation(cont'd)

ID	Point of view	Considerations	PoC	Dev.	Ops.
CE-8	Is there a high level of customer cooperation and involvement?	● Degree of customer cooperation and involvement in the development of the AI system ・Is the customer cooperative in providing and verifying the data held by the customer during [PoC][development]?	○	○	
		● Ongoing customer cooperation and involvement during operations ・Are you continuously exchanging opinions with your customers during operation? Are you getting feedback from the customer each time?			○

Explanation of CE-1

● Clarification of business issues to be solved by AI <PoC/Development>　<PoC>

Clarify the business issues using the following techniques.

Business issues should be identified quantitatively, such as the cost of market defects, in-process defects, and other losses, and the cost of inspection operations.

If they cannot be observed directly, break them down and consider whether they can be observed indirectly.

- Goal-oriented analysis
- marketing analysis
  - 4C analysis (customer value/customer cost/communication/utility)
  - 3C analysis (market/customers/competitors/company)
  - SWOT analysis (strengths/weaknesses/opportunities/threats)
  - STP analysis (market segmentation / target market / company's position)
  - PEST analysis (politics/economy/society/technology)
  - 4P analysis (product / price / distribution / promotion)
- 7 QC tools
  - Pareto charts, histograms, control charts, scatter plots, characteristic factor charts, check sheets, graphs

<Development>

- Monitor for changes in the content of the issue to be resolved.

Consequences of failing to pay attention to the notes

<PoC>

- I can't move from PoC to the next phase.
- Interruption of development. Interruption of development.

<Development>

- Increased development costs.
- The change in the task makes the system unable to achieve its purpose.

● The Potential of AI to Solve Business Problems<PoC/Development>

<PoC>

If it is difficult to set the goal that the customer is aiming for, consider separating the contract for [PoC] and [Development] in order to avoid risks for both the customer and the contractor.

In some cases, it is possible to solve the problem using conventional (and cheaper) methods without AI.

- Confirm whether the client's problem and the path to solving it are reasonable; consider whether the input data to and output data from the AI system are reasonable; consider whether the scale of the application and target is sufficient; and explain to the client the risks that may not be solved and obtain agreement. Evaluate literature and application examples to solve business problems, and explain them to customers.

<Development>

- Based on the results of the PoC, consider whether the problem can be solved without using AI. (As a result of collecting raw data and training data, there is a possibility that the problem can be solved even with a rule-based approach.)

Consequences of failing to pay attention to the notes

<PoC>

- Unable to move from PoC to the next phase.
- Development is suspended.

<Development>

- Increased development costs.
- change in the task will result in a system that fails to achieve its objectives.

● Effect of solving business problems to be solved by AI <PoC/Development>

<PoC>

- Define "how far do you want the system to go" while taking into account "cost effectiveness".
- Align the recognition of the level of automation expected by the introduction of the AI system and the level of automation expected by the customer to solve the business problem. (Reference: Coordination between Man and Machine in Automated Driving [3])

<Development>

- Monitoring for changes in cost effectiveness.
- Monitor the level of automation to see if it has changed.

Consequences of failing to pay attention to the notes

<PoC>

- You can't move from PoC to the next phase.
- Interruption of development.

<Development>

- Increase in development costs.
- change in the task will result in a system that fails to achieve its objectives.

● Understanding of continuous improvement for maintaining AI performance <PoC/Operation>
<PoC>

- Do you have a contract in place to maintain performance?
- Identify the work and costs required to maintain the performance of the AI. Does the cost of iterative relearning, maintaining and improving inference accuracy and environmental dependencies need to be considered?
- Clarify the criteria for maintaining the performance of AI. Is AI performance after development (at the time of introduction) used as the standard, and is performance maintenance and improvement judged based on that standard after operation?
- Clarify the metrics to be targeted in each phase for the expected effect of AI. For example, even if there is a target of "20% defect reduction", it is difficult to verify it using that as an indicator in PoC, so it is necessary to clarify the target metrics for each phase.
- Is there a defined mechanism for maintaining performance?

<Operation>

- Monitor whether the implementation of the AI is achieving the expected results.

Consequences of failing to pay attention to the notes
<Operation>

- increase in operational costs.
- cause the operation does not go well and the result is far from the expected one.

● Satisfaction with AI-Based Business Problem Solving <Operation>

- Conduct a customer satisfaction survey. Although the satisfaction survey will be conducted during operation, it should be conducted during [PoC] and [Development] to ensure that there is no divergence between business issues and activities.

Consequences of failing to pay attention to the notes
<Operation>

- The system will not meet the objectives. Customers will not be satisfied.
- cause of loss of opportunity for future orders.
- Increased operational costs.
- The system will not be operated properly and the results will be far from the expected results.

7.6.3 Explanation of CE-2

● Understanding of AI output with probabilistic behavior <PoC/Development/Operation>
<PoC>

- Explain to the customer and obtain their agreement that since this is a stochastic operation, there are concerns that
  - Making probabilistic decisions requires a large amount of training data sets, which are expensive to collect. In addition, the collected training datasets may not produce the expected results, which may result in wasting the collected training datasets.
- It can be a trial-and-error effort to adjust the input data and hyperparameters. Since there is no guarantee of gradual improvement as adjustments are made, a decision may need to be made

whether to continue with the adjustments or to discontinue the project.
- Explain to the customer and obtain their agreement that AI may judge a good product as a defective product or a defective product as a good product, and that countermeasures must be implemented throughout the system.

<Development>

- Explain to the customer that since this is a probabilistic behavior, there is some risk of the behavior if the input data differs even slightly from the test data and obtain their agreement.
- Explain to the customer and obtain their agreement on the specific cases where the AI misjudges between good and bad products.

<Operation>

- Explain to the customer and obtain their agreement that there is a possibility of unintended judgments that differ from human judgments and that the more complex the AI model, the more difficult it is to identify the cause.
- Explain to the customer and obtain their agreement that some decisions may be worse than the previous ones as a result of updating the AI model.

Consequences of failing to pay attention to the notes
<PoC / Development>

- For example, the percentage of correct inferences for xxxxx should be 100%.
- The development cost will increase as countermeasures against inference errors are required one after another.

<Operation>

- An unexpected problem can result in enormous damage.
- If an inference error occurs, we could be held liable for it.

● Risk tolerance of results output by probabilistic behavior <PoC/Development/Operation>
<PoC/Development>

- Verify whether the risk of output results (unexpectedness, errors in judgment/inference) due to stochastic behavior is acceptable and obtain agreement. If necessary, evaluate the risk of problems that may occur due to stochastic behavior and consider countermeasures. Depending on the contents of the countermeasures, they should be incorporated into the system implementation.
- Conduct a hazard analysis (using FMEA, etc.) of the overall system structure (clarifying the scope of AI's responsibility (level of automation)) and the case where AI makes a mistake, and verify whether the system is acceptable and obtain consensus. (Example) The following methods are used.
  - FMEA
  - FTA
  - HAZOP
- Identify the risk.
- Determine the priority level. (Priority = probability of occurrence x amount of damage)
- Develop risk countermeasures (hold, avoid, increase, share, eliminate risk sources, change likelihood, change consequences).

<Operation>

- Follow up on whether the risk measures taken during [PoC][Development] are working well.
- Formulate countermeasures (acceptance, avoidance, transfer, mitigation) in the event that risk countermeasures do not work and become apparent.
- The analysis procedure is the same as for [PoC] and [Development].
- During operation, risk analysis is performed because changes in the environment such as temperature may also have an impact.

Consequences of failing to pay attention to the notes
<PoC / Development>

- For example, the percentage of correct inferences for xxxxx must be 100%.
- The development cost will increase as countermeasures against inference errors are required one after another.

<Operation>

- An unexpected problem can cause enormous damage.
- If an inference error occurs, we could be held liable for it.

7.6.4 Explanation of CE-3

● Comprehension of agile software development <PoC/Development>
<PoC>
For more information on agile software development, please refer to the PA.

- Agree with the customer that the process will be developed in an agile manner against the goals set during PoC (e.g. percentage of correct answers).

At this time, it is also agreed what will be done if the target is not achieved within the period.

- Agree with the customer that it will be necessary to collect new training data and improve the quality of training data to achieve the target (e.g., correct answer rate) during PoC.
- Agree with the customer to increase the number of training data and improve the accuracy step by step by repeating trial and error during PoC.

<Development>
For more information about agile software development, please refer to PA.

- Agree with the customer on the goals during development as well as during PoC.
- agree with the customer on the release date of the AI component and its performance. Especially when the AI component and the surrounding development are done by different companies.
- Incorporate operational considerations into the AI system implementation. How to log, how to update the AI component, etc.

Consequences of failing to pay attention to the notes

- Repeated implementation does not improve the rate of correct answers and prevents the next step from being taken. This results in delays in planning and increased development costs.

● The level of understanding of continuous improvement for maintaining AI system performance <Operation>
● Understanding of verification methods for post-operation improvements <Operation>

- Training data for inference targets obtained during operation may deteriorate in inference performance (e.g., correct answer rate) due to aging, changes in the installation environment, and changes in public sentiment. For this reason, it is necessary to improve the AI model as appropriate.
- Verification of improvements during operation should be done with test data during operation.
- The following should be agreed upon for continuous improvement and validation of performance maintenance
  - Explain and agree on the operational improvements to the customer and sign the contract.
  - The handling of training data collection during the operation should also be specified in the contract.

Consequences of failing to pay attention to the notes
<PoC/Development>

- As a result of implementation, the target (e.g., correct answer rate) is not reached and development is suspended.
- Development is suspended due to increased cost of training dataset collection.
- Even after repeated implementation, the correct answer rate does not increase and the next step cannot be taken. This results in delays in planning and increased development costs.

<Operation

- If training data based on raw data at the customer site cannot be used for performance improvement, the requirement for performance improvement cannot be met.

7.6.5 Explanation of CE-4

● Understanding of the need for training data that is consistent with the customer's business challenges <PoC/Development>

- Help students understand that there needs to be a relationship between the business problem and the raw data collected.
  - Goal Oriented Analysis
  - Characteristic Factor Analysis
  - Statistically significant from scatter plots, correlation coefficient, etc. (uncorrelation test)

Consequences of failing to pay attention to the notes

- If sufficient training data is not available, it will be underserved relative to performance (especially in the percentage of correct answers).

● Understanding of the quality and quantity of training data required to train AI models <PoC/Development>
See DI-1 and DI-2.
Consequences of failing to pay attention to the notes

- If sufficient training data is not available, it will be underserved relative to performance (espe-

cially in the percentage of correct answers).

● Customer understanding of changes in input data trends during operation <PoC/Development/Operation>
<PoC/Development>

- Introduce the case study and have the customer understand that the trend of input data changes depending on 5M+E (people/equipment/method/material/inspection/measurement + environment).
- Organize the causal relationships among the changes in trends estimated by each explanatory variable and explain the possibilities. For example, changes in the installation environment, deterioration over time, etc.

<Operation>

- Establish a mechanism to monitor the input data content during operation to ensure that the input data is not different from the training data (discussion required).

Consequences of failing to pay attention to the notes
<PoC/Development/Operation>

- If sufficient training data is not available, it will be underserved relative to performance (especially in terms of correct answer rate).

<Operation>

- If inferences are made with input data of different trends during training and the results differ, disputes may arise.

Explanation of CE-5
● Contractual arrangements regarding intellectual property rights, including copyrights for AI systems,
- Level of understanding of intellectual property rights, including copyrights of AI systems,
● High security of customer data used in the AI system, clarification of the scope of information disclosure and handling restrictions
● Arrangements regarding the rights of data included in the AI system <PoC/Development>

There is a possibility that amendments to laws and regulations, changes in interpretation, etc. may change the scope of use of data, so it is necessary to check trends in legal amendments, etc. as appropriate. Although data does not give rise to ownership rights, it may give rise to rights such as copyrights (Article 30-4, Item 2 of the Copyright Act allows the use of data for learning purposes even if it is the work of a third party). In addition, there may be certain restrictions on the use of data due to contractual or legal (e.g., Personal Information Protection Law) regulations. It is necessary to consider the use of data appropriately in light of such restrictions, and to reach a certain agreement with the customer. In addition, it should be noted that foreign laws and regulations may apply when handling foreign data.

There is a possibility that the scope of data availability may change due to amendments to laws and regulations or changes in interpretation, etc. Therefore, it is necessary to confirm trends in legal amendments, etc. as appropriate.

Although data does not give rise to ownership rights, it may give rise to rights such as copyrights (Article 30-4, Item 2 of the Copyright Act allows the use of data for learning purposes even if it is the work of a third party). In addition, there may be certain restrictions on the use of data due to contractual or legal (e.g., Personal Information Protection Law) regulations. It is necessary to consider the use of data appropriately in light of such restrictions and to reach a certain agreement with the customer.

In addition, it should be noted that foreign laws and regulations may apply when handling foreign data.

- Define who owns the rights to the raw data and training dataset provided by the customer during PoC and the scope of such rights. For example
  - Are the raw and training datasets allowed to be used to improve the performance of the AI model after PoC?
  - Can the raw and training datasets be used in new proposal activities?
- Do you have permission from the provider to use the information as input data for inference? In addition, confirm that the output data resulting from the inference is ethically sound and obtain agreement with the customer.
- Share with the customer the understanding of the data management method, retention period and disposal method.
- Share the following perspectives to identify any issues and develop a policy to address them
  - Security: Ensure the robustness and reliability of the AI system.
  - Security Principle: To ensure that the AI system does not pose a risk to the life and physical safety of users or third parties.
  - Ethics Principle: Respect for human dignity and individual autonomy.
  - accountability principle: Accountability to users and other relevant stakeholders.

Consequences of failing to pay attention to the notes

- The following effects are possible.
  - cause of disputes due to infringement of third party's intellectual property rights
  - Disadvantages such as administrative guidance due to violation of the Personal Information Protection Law
  - cause of social criticism, possible brand loss, etc.

● Contractual arrangements regarding intellectual property rights, etc., including copyrights of the AI system improved after operation <Operation>

The scope of data availability may change due to amendments to laws and regulations or changes in interpretation, etc. Therefore, it is necessary to confirm trends in legal amendments, etc. as appropriate.

Although data does not give rise to ownership rights, it may give rise to copyrights and other rights (in accordance with Article 30-4, Item 2 of the Copyright Act, use of a third party's work for learning purposes may be permitted). In addition, there may be certain restrictions on the use of data due to contractual or legal (e.g., Personal Information Protection Law) regulations. It is necessary to consider the use of data appropriately in light of such restrictions and to reach a certain agreement with the customer.

In addition, it should be noted that foreign laws and regulations may apply when handling foreign data.

- Define the scope of the AI system and to whom the rights belong, including the updating of the AI model after release.
- Define who owns the rights to the data collected during operation and to the updated AI model and the scope of those rights. For example
  - Is it acceptable to view and process the input and output data to the AI component if it needs to be monitored during operation?
  - Who is responsible for updating the AI model? Who is responsible for the updated AI

model?
  - Share with the customer an understanding of how the data will be managed, retained and disposed of.

Consequences of failing to pay attention to the notes

- The following effects are possible.
  - cause of disputes due to infringement of third party's intellectual property rights
  - Disadvantages such as administrative guidance due to violation of the Personal Information Protection Law
  - cause of social criticism, possible brand loss, etc.

7.6.6 Explanation of CE-6

understanding of difficulties in explaining the AI model

● Understanding of Difficulties in Explaining AI Models and Accuracy <PoC/Development/Operation>

<PoC / Development>

- Create a trade-off table between performance and ease of explanation, and explain it to the customer. Show examples that are difficult to explain. (e.g. matrix formulas)
- Explain to the customer that there are cases where the process of reasoning with the created AI model is difficult to explain. Decide how to handle such cases.

<Operation>

- Monitor that the data required for the description is being collected.

Consequences of failing to pay attention to the notes

<PoC/Development/Operation>

- If there is no agreement on explainability, it may not be possible to respond to the customer's request for explanation. There is a possibility that the customer will not be satisfied with the answer.

● Comprehension of AI model selection <PoC/Development>

- Define the AI model options and rationale for selection and reach agreement with the customer.

Consequences of failing to pay attention to the notes

- If the AI model is not sufficiently effective due to the AI model, or if the AI model needs to be re-selected, this could lead to disputes over liability.

7.6.7 Explanation of CE-7

● Clarification of customers and stakeholders <PoC/Development/Operation>

It must be decided at the time of each contract: [PoC], [Development] and [Operation].

- Define the overall system structure (the scope of AI's responsibility and the connection to existing systems), identify stakeholders and clarify the degree of their involvement. If necessary, perform risk management. See CE-2.

Consequences of failing to pay attention to the notes
<PoC/Development/Operation>

- If a stakeholder is not identified, one or more of the following events may occur, resulting in rework and the cancellation of the PoC/Development/Operation.
  - Incorrect definition of the business problem.
  - Incorrect or inaccurate data to use.
  - The data you want to use is not available. etc.
- Lack of clarity on responsibility for the output of the AI system will lead to disputes.

● Clarifying Responsibility for AI System Outputs <PoC/Development/Operation>

- Define responsibilities and agree on them with relevant stakeholders. This includes responsibility for the input data to the AI component, responsibility for making decisions on the output data that is the result of inference, and responsibility for preparing the dataset for retraining.
- In addition, risk management should be performed for inferred events. See CE-2.
- If it is difficult to set quality goals for the AI system or to set conditions for the completion of PoC, separate PoC, development and operation contracts and clarify the goals of each contract in order to avoid risks for both the customer and development. For example, a Statement Of Work (SOW) should be created and agreed upon.

Consequences of failing to pay attention to the notes

- If a stakeholder is not identified, one or more of the following events could occur, resulting in rework and the cancellation of the PoC, development, or operation.
  - Incorrect definition of the business problem.
  - Incorrect or inaccurate data to use.
  - The data you want to use is not available. etc.
- Lack of clarity on responsibility for the output of the AI system will lead to disputes.

Explanation of CE-8

● The level of cooperation and involvement of the customer in the development of the AI system <PoC/Development>

<PoC>

- The purpose of each phase, the data that the customer needs to prepare (raw data or training data set) and the concerns should be described in the project plan and agreed upon with the customer before proceeding.
- Share the risks with the customer and agree on countermeasures against the failure to achieve the expected results or the inability to achieve the expected goals. Refer to CE-2 for risk analysis and countermeasure planning.
- Share the PoC objective (to determine the utility of the data provided by the customer) with the customer and obtain agreement.
- Confirm that the customer will continue to provide the data necessary for AI training. Continue to provide the quantity and quality of data recognized and combined in CE-4.

<Development>
Explain the following to the customer and obtain agreement.

- Confirm whether the actual operation data is within the expected range against the customer-

provided data (PoC experiment data).
- Confirm the impact of AI model changes when they occur.
- Confirm with the customer the procedure for updating the AI model during operation and the method of pre-checking. (e.g., release with results from customer-provided data)
- clearance Confirm that the customer data required for AI training will be provided on a continuous basis. (e.g., release with results from customer-provided data)

Consequences of failing to pay attention to the notes
<PoC>

- The following event(s) could result in rework or cancellation of the PoC:
  - Incorrect definition of PoC objectives.
  - Discrepancies in the perception of risk.
  - Inability to use the data you want to use.　　etc.

<Development>

- AI cannot be implemented on the target. Cannot achieve the desired performance. Start over from requirement definition and PoC.

● Ongoing customer cooperation and involvement during operation <Operation>

- Share the feedback method of operation results and the method to obtain exception data in advance. For urgent patch support, define the content (frequency, priority, etc.) and share it with the customer.
- Continue to provide the quantity and quality of data recognized in CE-4.

Consequences of failing to pay attention to the notes

- AI's effectiveness in addressing business challenges is attenuated due to reduced accuracy.

7.6.8 Data Integrity

Due to the environmental dependency of industrial systems, even closed systems such as factories are subject to numerous factors of data variability. No matter how rigorously change management is implemented, it is impossible to classify and verify all the data to be specified and handled during PoC and development against unexpected (inexperienced) variation factors such as natural deterioration of equipment, changes in procured materials, and changes in the natural environment and weather. Therefore, it is necessary to define the targets in advance in stages, and to check the variations and contents of the data.

Table 7.9　Considerations for Data Integrity

ID	Point of view	Considerations	PoC	Dev.	Ops.
DI-1	Sufficiency of the amount of training data	● Ensure the amount of data required for AI training ・Is the amount of data required for verification (PoC) for problem solving available?	○		
		・Is the amount of training data sets necessary for development for operation available?		○	
		● Securing data for cross-validation, generalization performance, etc. ・Is the amount of data enough to confirm not only training but also cross-validation and generalization performance?	○	○	
		● Evaluation of "bulking" data ・Has the method of bulking and the added data been evaluated for appropriateness?	○	○	
		● Evaluation of "bulking" of data using data obtained during operation ・Is it evaluated whether the assumptions made during development for "bulking" were appropriate for the distribution and labeling of additional data obtained during operation?			○
DI-2	Relevance of training data Conformance of training data to requirements	● Consistency of data used for training with business issues ・Is data that leads to solving issues provided by the customer? ・Is the data that leads to the solution of the problem provided by the customer, or is it possible to generate or acquire it?	○	○	
	Appropriateness of training data Complexity of training data	● Ensuring the quality of training data set ・Are sample data obtained exhaustively after defining the assumed population, and is there any bias?	○	○	
	Consideration of the nature of the training data Validity of the value range of the training data Legal compliance of the training data	● Characterization of the training data If the data have characteristics, have selection bias, information bias, and confounding issues/risks been assessed? Has the rationale for removal/correction of outliers and missing values and the method of action taken been based on policies such as acceptance and exclusion?	○	○	

Table 7.10　Considerations for Data Integrity (cont'd)

ID	Point of view	Considerations	PoC	Dev.	Ops.
DI-2(cont'd)		● Necessity of operational data ・Did you establish a system to record data that can only be collected in actual operation, because data that was not available during development may be obtained during operation? ・Have you ensured that the data corresponds to errors and diversity discovered in operation?			○
		● Characteristic evaluation of input data ・Does the data in operation have a bias different from that at the time of introduction? ・Does the data in operation have a bias that differs from that at the time of installation, and is the background analyzed? ・Is the rationale for removal and correction of outliers and missing values, and the method of action taken, based on policies such as acceptance and exclusion? Is it possible to assume system maintenance?			○
		● Complexity of the data definition (model to be assumed) for the business problem (phenomenon) ・When modeling the business problem, are the number of explanatory variables and causal relationships in the training data set too complex or too simple? Also, is multicollinearity taken into account?	○	○	
		● Appropriateness of acquisition route and management of training data set ・Is the acquisition/acquisition route of the training data set clear, and are there any deficiencies in the data management method?	○	○	
		● Are the labels and correct answers correctly attached? ・Is the data set appropriate for training, including the correct answers? (The correct values to be attached depend on the problem, e.g. labels for discrimination problems, values for regression problems, etc.)	○	○	

Table 7.11 Considerations for Data Integrity (cont'd)

ID	Point of view	Considerations	PoC	Dev.	Ops.
DI-3	Validity of data for verification	● Independence of training data set used for cross-validation, generalization performance, etc. ・Is the training data and test data used for cross-validation and generalization performance, etc. separated and managed independently?	○	○	
		Independence of training data sets used for cross-validation and generalization performance in retraining ・Is the training data and test data used for cross-validation and generalization performance in retraining and additional training separated and managed independently?			○
DI-4	Consideration of Influence of Online Learning	● Implementation of Learning Exclusion Mechanism for Outliers ・When performing online learning during operation and incrementally adding/replacing models, is a mechanism established to define reliable data intervals and prevent learning with unexpected data?		○	
		● Monitoring for outliers ・Do you patrol the quality of the input data, for example, by monitoring whether the data to update the model is out of the expected data interval?			○
DI-5	Validity of Data Processing Programs	● Validation of Programs Used for Data Preprocessing, etc. ・Does the supplier confirm that the programs used for data processing are valid? ・Do you confirm that the program for data processing is appropriate, and do you have the results of the confirmation and can you show them?	○	○	

7.6.9 Explanation of DI-1

● Securing the amount of data required for AI training <PoC/Development>

In general, the more data used for training, the closer it is to the population. However, depending on the training, no matter how much the amount of training data is increased, the performance may become saturated and the verification efficiency may decrease. For this reason, in the PoC phase, it is also important to evaluate the amount of data required to obtain the expected learning performance in order to control the cost and time required for verification in the development phase. In addition, the algorithm varies depending on the problem to be solved, and the amount of learning data required for the algorithm also varies (Reference: scikit-learn algorithm cheat sheet [1]).

Consequences of failing to pay attention to the notes

If the training dataset is too small, the expected model cannot be built and the correct classification and regression results cannot be obtained during operation. On the other hand, if there is too much data for training, it will take more time to train and the recording capacity will be required in proportion to

the amount of data.

● Secure data for cross-validation, generalization performance, etc. <PoC/Development>
In many cases, not all the data can be used for training because the training dataset is used separately for training and testing. Therefore, it is necessary to ensure that there is an adequate amount of data to build the best model.

Consequences of failing to pay attention to the notes
As in the case of a small amount of data for training, the expected model cannot be built and the correct classification and recurrence results cannot be obtained during operation.

● Evaluation against "bulky" data <PoC/Development>
When the training data set for PoC is insufficient, the amount of data may be increased by bulking. At this point, it is important to confirm the validity of the bulking method and the increased data.

Consequences of failing to pay attention to the notes
Generating data with trends that differ from the expected population by bulking can lead to poor generalization performance.

● Evaluation of "bulking" of data using data obtained during operations When the training data set is small, the data may be slightly processed to "bulk up" the data. Bulking is a process of shifting, inverting, or adding noise to the data. We examine whether it is possible to construct a model with this kind of bulking data. For example, in the case of an image inspection system, it is recommended to assume that there are factors that affect the input data to the AI, such as the state of the camera and the surrounding conditions, in addition to the sample images of good and bad products. The following techniques can be used for bulking up. Image: Data Augmentation, Signal: various noise additions

Consequences of failing to pay attention to the notes
Applying the wrong bulking method can result in poor generalization performance. Even if the "bulking" data is appropriate, additional training may not necessarily improve performance, in which case, evaluation by retraining should be considered.

7.6.10 Explanation of DI-2

● Alignment of data used for learning with business issues <PoC/Development>
When adding or changing data to improve the quality of training data, it is often a trial-and-error approach, so it is important to manage the change history so that it can be rolled back or reproduced. (See also PA-3) It is difficult to define "data that leads to the solution of a problem"; a realistic approach is to identify data that can solve the "problem" through trial-and-error by repeating learning and evaluation for the "problem to be solved," which should be clarified when considering the use of AI.

Consequences of failing to pay attention to the notes
If it is not present in the training data, the decision becomes difficult.

Ensuring the quality of learning data sets <PoC/Development>
To build a highly accurate model, it is recommended to treat not only positive examples but also

negative examples as training data. If there are many outliers, missing values, fluctuations, duplicates, etc. in the training data, it is highly possible that the model cannot be learned correctly, so preprocessing (cleansing) of the training data is necessary. Similarly, it is recommended to consider logarithmization and normalization of numerical values to normalize the bias.

Consequences of failing to pay attention to the notes

If there are few negative examples or outliers, the accuracy of the model will be reduced.

Characterization of training data <PoC/Development>

In most cases, it is not possible to have a training dataset with all of the population as input. Therefore, we assume a population and use the sample data in it. If there is bias in the sample data, the learned model is also affected by the bias*. Use sample data that adequately represent the assumed population. (*Data bias can be checked using histograms, principal component analysis, t-SNE analysis, scatter plots, etc.)

Consequences of failing to pay attention to the notes

If there is a bias in the training data, the model will have a lot of untrained data, which may lead to poor generalization performance.

Necessity of operational data <Operation>

For input data that has not been learned, the output cannot be defined. For example, if part D is input to a system that discriminates part A, part B, and part C, it cannot predict whether the output will be A, B, or C. Even when part A is input, if there is no training data for the side or back, A may not be output correctly. Therefore, it is important for the training data to have comprehensiveness against the input data during operation. However, it is practically difficult to have complete comprehensiveness, and untrained input data may be input during operation, so we also consider how to deal with such cases. Also, you should consider defining an Unknown class that does not fit into any of the above categories.

Consequences of failing to pay attention to the notes

If it is not present in the training data, the decision becomes difficult.

Characteristic evaluation of input data <Operation>

As the operation proceeds, the trend of input data may become different from when it was trained. In such a case, it is highly likely that the expected results cannot be obtained without updating the model.

Consequences of failing to pay attention to the notes

As the operation proceeds, the trend of input data may become different from when it was trained. In such a case, it is highly likely that the expected results cannot be obtained without updating the model.

● Complexity of data definitions (models to be assumed) for business issues (phenomena) <PoC/Development>

When there are many types of data (factors) used for learning, there is a possibility that factors with small influence on the learning results or data with high correlation among factors are included. If training is performed using such data, the amount of data used for training will increase and the verification cost will also increase. Therefore, at the time of PoC, we use simulations and multivariate analysis to examine factors that have a high impact on the learning results to narrow down the data and consider reducing the verification cost.

Consequences of failing to pay attention to the notes

If all the factors are used, the learning time may increase and resources such as memory may be used too much. It also takes time for verification.

Appropriateness of acquisition route and management of data set for learning <PoC/Development>

It is also necessary to pay attention to the acquisition and management of learning data. This includes ensuring that the data was not obtained without the customer's consent and that privacy-related data is properly managed.

Consequences of failing to pay attention to the notes

Improperly obtained or managed training datasets can lead to social and ethical issues.

● Is the label and correct answer value attached correctly <PoC/Development>

When a correct answer label is assigned to a training data set, it must be a correct label. If the label is incorrect, it will lead to an error in the model.

Consequences of failing to pay attention to the notes

Generalization performance will also be poor if there are many errors in the labels.

7.6.11 Explanation of DI-3

● Independence of training datasets used for cross-validation, generalization performance, etc. <PoC/Development>

In machine learning, if the number or variation of training data sets is small or the number of training times for one training data is too long, the model may fall into overlearning. In order to properly evaluate overlearning, the data used for training and the data used for cross-validation and generalization performance should be completely separated and managed. In order to separate the data, it is also important to clearly define how the data is managed.

Consequences of failing to pay attention to the notes

If the data management methods are not appropriate, training and validation data will be mixed and the generalization capability of the model cannot be properly assessed.

● Independence of training data sets used for cross-validation, generalization performance, etc. in re-training <Operation>

Training data and validation data should be appropriately separated in retraining/additional training.

Consequences of failing to pay attention to the notes

If the data is not managed in an appropriate manner, the training and validation data will be mixed and the generalization capability of the model cannot be properly assessed.

7.6.12 Explanation of DI-4

● Implementation of a learning exclusion mechanism for outliers <Development>

When online learning, in which data is updated one at a time, is incorporated in operation, it is

necessary to define expected data intervals and build a mechanism such as not learning unexpected data in order to prevent the model from being degraded by abnormal data. At the same time, a mechanism for checking data trends should be provided so that it is possible to record what kind of data was used to update the model.

Consequences of failing to pay attention to the notes

Without a mechanism, even if there is a large amount of anomalous data during operation, the model will be updated with the anomalous data without noticing the anomalous data.

● Monitoring for outliers <Operation>

When updating the model during operation, monitor the trend of the data used for updating. The trend of the data may change, or abnormal data may be input due to noise or other reasons. In such cases, the service user will be notified and will be able to choose whether to continue learning or to stop.

Consequences of failing to pay attention to the notes

If there is a large amount of anomalous data during operation, the model may be degraded by the data.

7.6.13 Explanation of DI-5

● Validation of programs used for data preprocessing, etc.

A program that pre-processes, creates and pastes data may be used, and it should be confirmed that this program works correctly. In general, preprocessing programs are rule-based programs, so the validity of these programs should be confirmed using tests that are performed in general software development. In addition, the result of the confirmation should be recorded so that it can be confirmed later.

Consequences of failing to pay attention to the notes

The wrong training dataset will be used when training the AI model and it will not train correctly. If the validity of the data processing program is not checked beforehand, it will be difficult to isolate the cause when the model has unintended behavior or performance.

7.6.14 Model Robustness

Due to the constraints described in the Data Integrity section, it is difficult to mathematically guarantee the completeness required for evaluating the robustness of the model using only the collected data. Therefore, it is necessary to collect data under various collection conditions (i.e., diversity of disturbances and parameters that the system must be equipped with), and to conduct training and robustness evaluation using directly used or processed data. The validity of using direct data or processed data is evaluated based on the system quality. (For example, when data variability due to disturbance is obtained by processing, the rigor of the demonstration of the processing method is determined based on the extent and risk of the impact of the processing result on the system quality.

Table 7.12　Considerations for Model Robustness

ID	Point of view	Considerations	PoC	Dev.	Ops.
MR-1	Sufficiency of model accuracy ・Are the values of evaluation indices for inference performance, such as correct answer rate, fit rate, recall rate, and F-value, sufficient for the requirements	● Definition of tentative specification of index values ・Do you explain the values of indices such as correct answer rate and fit rate to the customer? ・Are the customer's requirements and the optimal indicator values sorted out by the end of the PoC? ・Is it possible to hear the customer's requirements for indicator values (e.g., correct response rate of ****% or more) at the time of PoC? ・Is it possible to organize the customer's requirements by the end of PoC?	○		

Table 7.13　Considerations for Model Robustness (cont'd)

ID	Point of view	Considerations	PoC	Dev.	Ops.
MR-1 (cont'd)		● Validity of Learning Results ・Are the residuals of the correct answer rate and loss function after learning sufficiently converged? ・Do the rate of fit, rate of reproduction, and F-value reach the target?		○	
		● Reasonableness of AI system operation after operation ・Are factors that affect performance extracted after operation, and are performance targets set with a margin?			○
MR-2	Sufficiency of model generalization performance, sufficiency of model evaluation, sufficiency of model validation - Is generalization performance ensured? - Have appropriate indicators other than accuracy (such as AUROC) been selected and sufficiently evaluated to express the goodness of the model? ・Have you conducted sufficient cross-validation, etc.	● Investigation of generalization performance ・What generalization performance measurements are appropriate, and are they discussed and consistent with the customer?	○		
		● Targets for generalization performance ・Are the targets for generalization performance clearly defined? ・Is the generalization performance of the AI model after training significantly degraded compared to the correct answer rate at the time of training?		○	
		● Method for measuring generalization performance ・Have you determined the method for measuring generalization performance? ・When using cross-validation, do you ensure the variation of the training dataset to be used?		○	
		● Define the cross-validation method in operation ・Is the method of cross-validation determined so that it can be verified even when the variation of the training data set increases?			○
MR-3	Validity of Learning Process ・Did learning proceed appropriately ・Did not fall into local optimum	● Validity of Learning Process ・Are the correct answer rate and residuals of the loss function after learning sufficiently converged? ・Do the percentages of correct answers and residuals of the loss function in the learning process show any abnormal changes?	○	○	
		● Relevance of the learning process at relearning ・Are the correct answer rate and residuals of the loss function sufficiently converged after learning? ・Do the correct answer rate and residuals of the loss function in the learning process show any abnormal changes?			○

Table 7.14　Considerations for Model Robustness (cont'd)

ID	Point of view	Considerations	PoC	Dev.	Ops.
MR-4	Relevance of Model Structure ・Has consideration been given to whether the algorithm is appropriate ・Have you examined whether the hyperparameters are appropriate or not	● Relevance of AI Model Structure ・Is the rationale for selecting the selected AI algorithm and distillation presence/absence, and the rationale for setting hyperparameters clear? Is the rationale for selecting the algorithm explained and agreed upon with the customer?	○	○	
		● Record of hyperparameters ・Do you keep a record of what hyperparameters were set and verified? ・Do you keep records of what hyperparameters were set and verified, and are you able to explain the differences in AI model performance for each hyperparameter to customers? ・Does the customer understand that the hyperparameter settings affect the performance of the AI model?	○	○	
		● Recording of hyperparameters at the time of relearning ・Does the system keep a record of what hyperparameters were set for relearning?			○
MR-5	Robustness of the model ・Is the model robust to noise? Specifically, are error factors selected and their impact analyzed?	● Identify noise that affects AI models ・Do you identify noise candidates that affect the AI? Are noise candidates that affect AI identified? Specifically, are error factors selected and their effects analyzed? Specifically, are error factors selected and their impact analyzed?	○		
		● Relevance of noise tolerance (robustness) of AI model ・Does the performance of the AI model deteriorate significantly due to noise candidates?		○	
MR-6	Sufficiency of verification for model updates, consideration of model obsolescence	● Tolerance of performance degradation in relearning ・Is the degradation against the performance before relearning acceptable as a result of relearning due to changes in the characteristics of training data or addition of outputs?			○
		● Sufficiency of the inspection content for automatic updating and deployment of AI models ・When updating AI models automatically instead of manually, is it possible to sufficiently inspect that the changes in AI model characteristics and performance are acceptable?			○
MR-7	Appropriateness of the model as a program	● Testing External Libraries ・When evaluating the system, are unit tests and system tests against external libraries conducted?		○	
		● Scope of responsibility for external libraries ・Is the scope of responsibility for defects clarified with the library supplier?	○		

7.6.15 Explanation of MR-1

Definition of provisional specification of index value <PoC>

- Do you explain the definition of indicator values to customers?

Consequences of failing to pay attention to the notes

- Inconsistency occurs between the performance of the developed AI system and the customer's requirements.

● Validation of learning results <Development>

- Prepare a system to calculate accuracy indicators (fit rate, reproduction rate, F-value) during training. Does the created AI system ensure the required accuracy?

Consequences of failing to pay attention to the notes

- Looking too much or missing something can create an AI system that doesn't meet the customer's requirements.

● Validity of AI system operation after operation <Operation>

- Is the system operated to confirm the adequacy of the AI system by periodically monitoring changes in various characteristics and recalculating the accuracy index each time?

Consequences of failing to pay attention to the notes

- AI systems continue to operate without realizing that their performance is deteriorating.

7.6.16 Explanation of MR-2

● Investigation of generalization performance <PoC>

- Do you explain to customers what generalization performance is?
- Explain to the customer the methods to improve generalization performance (examples are provided below) and their characteristics.
  1. Regularization
  2. Cross-validation, etc.

● Generalization performance goals <Development>

- Are you able to explain the generalization performance verified in the PoC to the customer?
- Then, have you clearly defined the generalization performance as a requirement specification?

Consequences of failing to pay attention to the notes

- AI models may be operated in an overtrained state without sufficient generalization performance.

● Methods for measuring generalization performance <Development>

- In cross-verification, is the verification performed with sufficient test data?

Consequences of failing to pay attention to the notes

- AI models may be operated in an overtrained state without sufficient generalization performance.

● Define the cross-verification method for operation <Operation>

- Do you define a flow for periodically performing cross-validation and updating AI models? Has it been explained to the customer?

Consequences of failing to pay attention to the notes

- Continue to operate the AI model without noticing that its performance is deteriorating.

7.6.17 Explanation of MR-3

● Relevance of the learning process <PoC/Development>
Validity of the learning process for relearning <Operation>

- Is it possible to prepare a training data set that allows for sufficient convergence of learning regardless of whether it is during PoC, development or operation? Also, do you perform cross-validation using the training data set?
- In the learning process where the correct response rate and loss function are improved by iterative calculation, are the residuals of the correct response rate and loss function visualized, and is it confirmed that there are no abnormal changes and convergence is achieved?

Consequences of failing to pay attention to the notes

- The performance of the AI model when it is actually developed → operated differs from the performance during PoC.
- AI models will be created that depend on the training data used during training (overlearning).

7.6.18 Explanation of MR-4

● Relevance of the AI model structure <PoC/Development>
● Recording of hyperparameters <PoC/Development>
● Recording of hyperparameters during retraining <Operation>
<PoC/Development>

- Is the rationale for AI algorithm selection and hyper-parameter settings clear and approved through internal design review or other means?Is the rationale for AI algorithms and hyper-parameters understandable to the customer? The rationale for the AI algorithm and hyperparameter settings should be understandable to the customer, and the results of test data analysis should be presented. Since hyperparameter records are related to configuration management, please refer to PA-3.

<Operation>

- Is the rationale for AI algorithm selection and hyperparameter settings clear and approved by means such as internal design reviews? The rationale for the AI algorithm and hyperparameter settings should be understandable to the customer, and the results of test data analysis should be presented. Since hyperparameter records are related to configuration management, please refer to PA-3.

Consequences of failing to pay attention to the notes
<PoC>

- The performance of the PoC will be different from the performance when the AI model is actually developed and operated.

<Development>

- The results inferred by the AI component will be less reliable.

<Operation>

- When AI model degradation is suspected, it is not possible to determine whether it is due to a change in hyperparameters or a difference in the environment from the time of training.

7.6.19 Explanation of MR-5

● Identification of noise affecting AI models <PoC>

- Do you explain the robustness to noise to your customers? The following two types of noise are considered here.
  1. Noise generated in time series data or image data from sensors or actuators
  2. The inclusion of data outside the expected environment.
- Have the above noise candidates been extracted in advance?

Consequences of failing to pay attention to the notes

- The performance of the AI model when it is actually developed → operated differs from the performance during PoC.

● Relevance of noise tolerance (robustness) of AI models <Development>

- Are the specifications for noise filter removal of raw data confirmed? (For example, if it is sensor data, is signal analysis such as low-pass/high-pass filter or FFT performed as preprocessing?
- Are data outside the assumed environment excluded from the training data set when training?

Consequences of failing to pay attention to the notes

- The AI component's correct answer rate is reduced due to the introduced noise.

7.6.20 Explanation of MR-6

● Tolerance of performance degradation in relearning <Operation>

- Is it possible to set the acceptable range of degradation and its KPI in consideration of business issues?
- Is it possible to assume the range of changes in the characteristics of training data and additional output?
- Are the results of re-training/additional training due to the assumed changes in training data characteristics and additional outputs within the acceptable range of degradation?

Consequences of failing to pay attention to the notes

- Increased cost of addressing post-operational degradation due to lack of consideration of degradation.

● Sufficiency of checks for automatic AI model updates and deployment <Operation>

- Are sufficient inspection and testing automations in AI model retraining/additional training implemented?
- Are metrics to measure changes in characteristics and performance clear and are actions to be taken if they exceed the acceptable range determined?

Consequences of failing to pay attention to the notes

- Unexpected AI model degradation can occur unknowingly.

7.6.21 Explanation of MR-7

● Testing external libraries <Development>

- When evaluating AI systems, unit tests and system test specifications based on the API specifications of external libraries shall be created and evaluated. The data used for the test should be equivalent to the production operation, but if such data cannot be obtained, dummy data for the test may be substituted.

Consequences of failing to pay attention to the notes

- The risk of system quality degradation occurs when an AI system is released without evaluation of external libraries.

● Responsibility of external libraries <PoC>

- An agreement with the vendor of the external library (including subcontractors) on the scope of responsibility for any modifications and the costs and man-hours required in the event of a failure.
- If the software is free software, check the license type such as GPL or MIT to determine if it can be incorporated into the AI system.

Consequences of failing to pay attention to the notes

- Defects in external libraries will take time to be fixed, delaying the development of the AI system.
- It will be necessary to consider workarounds in the event of defects that deviate from the coverage of the free license.

7.6.22 System Quality

While it is difficult to guarantee Model Robustness and Data Integrity, industrial systems have high requirements for stable operation, safety, and security of the system due to their social roles. Depending on the application and business entity, there is a necessity to account for certification by external organizations and quality assurance of the entire system through quality assurance processes. There, the quality impact of adding AI components on the overall system and explanations for environmental dependencies such as aging are explained to the stakeholders of the industrial system.

Table 7.15　Considerations for System Quality

ID	Point of view	Considerations	PoC	Dev.	Ops.
SQ-1	High Customer Expectations Whether the target is "human-like"	● Consideration of Customer Value for AI System ・Is the AI system provided suitable for solving the customer's business issues? Also, is it possible to measure its effectiveness and value?	○	○	○
SQ-2	Are the fatalities of quality incidents that may occur kept acceptably low ・The fatalities of quality incidents vary depending on the domain (harm to body or life, economic damage/impact on society and environment/comfort, unattractiveness, lack of meaning, unethical) AI	● Extraction of risks to the AI system ・A risk analysis (e.g., FMEA, situation analysis, HAZOP analysis, etc.) is conducted, taking into account that the output of the AI system is a stochastic operation, to identify risks that may occur when using the AI system	○	○	
		● Consideration of risk prediction for AI system ・Are risk analyses (e.g. FMEA, situation analysis, HAZOP analysis, etc.) and reduction measures for quality incidents caused by the output of the AI system considered, taking into account that the output of the AI system is a stochastic behavior?		○	
		● Review/addition of risk prediction for AI systems ・Is the risk analysis (situation analysis, HAZOP analysis, etc.) conducted during development reviewed at the operational stage, or is it added when new risks emerge?			○

Table 7.16　Considerations for System Quality (cont'd)

ID	Point of view	Considerations	PoC	Dev.	Ops.
SQ-3	Are the system's accident reachability, safety functions, and attack resistance sufficient ・Goodness and number of protection mechanisms ・Avoidability and controllability ・Self-healing Is the contribution of AI to the system controlled	● Ensuring Safety for AI Systems ・Is the architectural design, including safety and security mechanisms, considered? ・Is the safety design for the output of the provided AI system considered?		○	
		● Secure control mechanisms to prevent abnormal output ・Is the system capable of self-judgment of AI abnormality and of monitoring output data and controlling it to an appropriate output range implemented? Is it implemented?		○	
		● Is the system maintained (ability to detect, diagnose, and repair failures and abnormalities)? ・Is the system implemented to detect the degradation of AI reliability and seamlessly transition to a non-AI system? (A mechanism to stop AI without stopping the system).		○	
		● Security of input data to be fed back for relearning ・Is it possible to prevent malicious data that leads to performance degradation from being mixed into the training data used for learning? Or, is there a mechanism to eliminate malicious data before training?		○	
		● Ensuring the safety of input data during operation ・Is there a mechanism implemented to detect and eliminate data that may lead to abnormal behavior or have malicious intent regarding input data during operation?		○	
		● Monitoring the security of input data fed back to relearning ・Is it possible to prevent the inclusion of malicious data that leads to performance degradation in the data fed back to learning? Or, is there a mechanism to eliminate malicious data before training? Does the system have a mechanism to eliminate abnormal or malicious data that may lead to performance degradation?			○
		● Monitoring the safety of input data during operation ・Is there a mechanism to eliminate abnormal or malicious data that may lead to abnormal behavior for input data used for inference during operation?			○

Table 7.17　Considerations for System Quality (cont'd)

ID	Point of view	Considerations	PoC	Dev.	Ops.
SQ-4	Can the frequency of occurrence of events that may cause quality incidents be estimated to be low , Frequency of occurrence of events , Coverage of events , Controllability of occurrence of events	● Assessment of the adequacy of the safe operation of AI systems Can the safety of the AI system be demonstrated using statistical methods, etc. based on the actual operation of the AI system?		○	○
SQ-5	Are assurance, accountability and acceptability to stakeholders sufficient?	● Explainability of the AI system ・Is it possible to explain the rationale for the AI output results, or show the validity of the results using statistical methods, etc.?	○	○	○
		● Convincibility of the AI system ・Was the customer convinced by the rationale for the output results of the AI system?	○		
		● Ease of understanding of AI system ・Were customers convinced by the rationale for the output results of the AI system?			○
SQ-6	Does the introduction or change of AI adversely affect the quality of the overall system behavior, performance, etc.	● Relevance of system behavior during operation ・Has the possibility of degradation of the system performance or other quality due to continuous operation been considered? ・Is there a mechanism to check for changes in the characteristics of AI system input data or degradation in performance? ・Is it possible to envision a method for maintaining normal system operation in response to changes in the characteristics of AI system input data, with the aim of preventing quality incidents? ・Has the system been evaluated as a whole and in meaningful subsystem units?		○	○
SQ-7	Can the system be expanded for future increases in data and processing volume?	● Assumption of the amount of input/output data to be collected during operation ・Is the amount of input/output data and the amount to be accumulated during operation estimated and reflected in the system requirements? If there is input/output data that is not currently acquired but is expected to be acquired in the future, is it considered as an expansion possibility?	○	○	
		● Monitoring and control of the amount of input/output data during operation ・Does the system monitor the amount of input/output data accumulated during operation and delete it based on the standard? In addition, is the system monitored to ensure that the established estimate is not exceeded?			○

Table 7.18　Considerations for System Quality (cont'd)

ID	Point of view	Considerations	PoC	Dev.	Ops.
SQ-8	Do you plan for system configuration items (hardware and OSS)	● Consider various constraints on software and hardware * Consider whether the performance required by the customer can be achieved when implementing AI as a target. ・Are there any hardware limitations for the implementation target? ・Is there a need to reduce the size of the training dataset or trained model? ・Is there a need to reduce the size of the training dataset or the trained model, and to what extent is performance degradation allowed? ・Have you considered how to distribute the retrained models? ・Consider the necessity of retraining the model during operation.	○		
		● Formulation of hardware assuming operation ・Is the hardware selected according to the load and amount of data during operation? ・Is the hardware selected according to the load and amount of data during operation, and is maintenance and failure support planned?		○	
		● Selection of software in consideration of updates ・Do you use software in consideration of the update frequency and support period of various software such as OS and OSS? Have you decided how to respond to software updates and what to do when support ends?		○	
		● Hardware constraints that depend on the performance and structure of the model ・ Trade-off between the size of the network of inference programs and the required memory.		○	
		● Hardware maintenance based on the plan ・During operation, are the load and the amount of data as expected? ・Do you maintain the hardware based on the plan during development?			○
		● Software Update ・Does the company update the system for software updates such as OS and OSS, especially when there are security updates?			○
		● Agree on the operational design of various constraints of software and hardware ・Does the system have an operational design for the method of relearning and the method of delivering relearning, and obtain agreement with the customer?			○

7.6.23 Explanation of SQ-1

● Consideration of customer value for AI systems <PoC/Development/Operation>

<PoC>

- Have you clarified the customer's business issues and shared your understanding with the customer?
- Have you and the customer agreed on the necessity of AI in solving the customer's business challenges?

<Development>

- Have you defined quality objectives to determine that you have solved the customer's business problem?
- Have you identified the constraints of using AI in the customer's business environment (e.g., the number and quality of available training data, conditions for handling training datasets, and the need for redundant configurations to prevent AI system outages)?
- Has the system been evaluated against the target quality in system testing to confirm that it is suitable for solving the customer's business issues?
- If value is difficult to measure, is the relationship to alternative metrics that can be measured reasonable?

<Operation>

- Have you confirmed that the AI system achieves the quality targets set during development and solves the business issues of customers?
- Is there a mechanism to continuously check and provide feedback on whether the AI system is solving the quality objectives set during development and the customer's business issues?

Consequences of failing to pay attention to the notes

- Achieving quality goals at design time without consideration of customer value may not solve the customer's business problem.
- Value may degrade over time, so even if it is achieved at release time, the business problem may not be solved over time.

7.6.24 Explanation of SQ-2

● Identification of risks to AI systems <PoC>

- Given that the output of the AI system is a stochastic behavior and unexpected output results may be obtained, did you conduct risk analysis (e.g., FMEA, situation analysis, HAZOP analysis, etc.) to identify risks that may occur when using the AI system (economic risk, safety risk, environmental risk)? (economic risk, safety risk, environmental risk)?
- Have you covered the risks that may occur during the use/operation of AI systems?
- Have you considered the stochastic behavior specific to AI systems and the low explainability of output results in your risk analysis?
- Is the severity of the extracted risks appropriate?

Consequences of failing to pay attention to the notes

- If the extracted risks are not appropriate (low comprehensiveness or incorrect severity), appropriate countermeasures may not be reflected in the architecture and design, resulting in an AI system that is vulnerable to unexpected risks. This can result in AI system shutdowns, the production of defective products or missed products due to the inability to detect abnormalities in AI system operation, or the possibility of danger to humans due to abnormal equipment operation.

● Consideration of risk prediction for AI systems <Development>

- Given that the output of an AI system is a stochastic behavior and may produce unexpected output results, have you conducted a risk analysis (e.g., FMEA, situation analysis, HAZOP analysis, etc.) to cover the risks that may occur during system operation (economic risk, safety risk, environmental (e.g. economic risk, safety risk, environmental risk) that may occur during system operation, and analyzed their severity?
- Are risk reduction measures (e.g., in-process monitoring, redundancy, implementation of rules for output assurance) designed based on the risk analysis?
- For the risks identified by the risk analysis, has the risk reduction method confirmed that the risks have been reduced based on the actual operation of the AI component?

Consequences of failing to pay attention to the notes

- If the extracted risks are not appropriate (low comprehensiveness or incorrect severity), appropriate countermeasures may not be reflected in the architecture and design, resulting in an AI system that is vulnerable to unexpected risks. This can result in AI system shutdowns, the production of defective products or missed products due to the inability to detect abnormalities in AI system operation, or the possibility of danger to humans due to abnormal equipment operation.

● Review and add risk prediction for AI systems <Operation>

- In addition to the risk analysis conducted during development (e.g., FMEA, situation analysis, HAZOP analysis, etc.), did you add any unexpected degradation of the output data of the AI system or the accuracy of the trained model due to stochastic behavior of the AI during operation as new risks?
- If there is a difference between the training data set used during development and the input data obtained during operation, the risk of accuracy degradation of the trained model increases. If there are differences, have you taken measures to reduce the risk by providing feedback to those in charge of AI system operation?

Consequences of failing to pay attention to the notes

- Inadequate risk analysis prevents a quick response when a risk occurs or proper prioritization when multiple risks occur. As a result, accidents and damages may occur.
- If risks found in the operational environment that were not assumed at the time of learning are not added, it is not possible to identify risks caused by changes in the characteristics of the operational environment or the external environment conditions surrounding the system, and failures or accidents may occur. This may result in AI system stoppages, the production of defective products or missed products due to the inability to detect abnormalities in AI system operation, or the possibility of danger to humans due to abnormal equipment operation.

7.6.25 Explanation of SQ-3

● Ensuring Safety for AI Systems <Development>

- In light of the fact that the output of an AI system is a stochastic operation and may produce unexpected output results, in order to ensure the quality of the AI system even in the event of safety risk events identified in the risk analysis, the design of fail-safe functions and rollback design of the system shall be considered.
- Have you considered the safety and security mechanisms in the design of the system?
- Considering the possibility that malicious data may be mixed in with the input data during operation, did you compare and analyze the correlation between the training data and the input data during operation using statistics such as the mean, variance, and correlation coefficient of the data, and test whether the data are sampled from the same population?
- If you were able to determine that the input data during operation were obtained from a different population than the training data during training, did you investigate the reason for the difference (system crack, change in installation environment, etc.)?

Consequences of failing to pay attention to the notes

- If fail-safe and rollback functions are not properly designed from a safety perspective, it may not be possible to maintain appropriate system output in the event of a risk.

● Securing a control mechanism to prevent abnormal output <Development>

- Is a mechanism implemented to monitor the output data and control the output to an appropriate output range even if it is abnormal output?
- Is a statistical method (outlier detection, change point detection, etc.) or a method using machine learning (k-nearest neighbor method, simple Bayesian method, etc.) used to detect abnormal output?

Consequences of failing to pay attention to the notes

- Without a control function to prevent abnormal output, the output value may remain abnormal in the event of a risk, reducing the safety of the system.
- If the ability to detect, diagnose, and repair failures and abnormalities is not ensured, the degradation of AI reliability due to risk occurrence may not be detected, or even if detected, appropriate action may not be taken. As a result, AI system shutdowns, the production of defective products or missed products due to the failure to detect abnormalities in AI system operation, or abnormal equipment operation may pose a risk to humans.

● Ensure maintainability (the ability to detect, diagnose and repair failures and abnormalities) for AI systems <Development>

- Has a mechanism been implemented to monitor AI components for failures and abnormalities and seamlessly transition to an alternative system that does not use AI when an abnormality is detected? (A mechanism to stop the trained AI model without stopping the system)

Consequences of failing to pay attention to the notes

- If the ability to detect and diagnose malfunctions and abnormalities is low, it may not be possible to prevent malfunctions and accidents from occurring. It may also create physical and economic risks to users. This may result in AI system stoppages, the production of defective products or

missed products due to the inability to detect abnormalities in the operation of the AI system, or the possibility of danger to humans due to abnormal equipment operation.

● Ensure safety to the input data fed back to relearning <Development>

- For the input data to be fed back for relearning during operation, implement a mechanism to prevent the inclusion of malicious data that may lead to performance degradation (e.g., operational input data obtained from a different population than the training data, outliers of the input data, etc.) or a mechanism to eliminate malicious data from the training data before relearning. (e.g., input data from a different population than the one used for relearning, outliers of input data, etc.), or a mechanism to eliminate malicious data from the training data before relearning? The mechanism to prevent the input of malicious data includes, for example, a method to keep the data secret so that the correct format of the input data cannot be known, and a method to make it easy to detect malicious data that has been mixed in after the data has been kept secret.
- If the malicious data cannot be eliminated for the training data before training, can the AI system be returned to the state before training with the training data contaminated with malicious data?
- After an anomaly in the AI system output data has occurred, has a mechanism been established to identify the malicious input data that has been mixed in, for example by retaining the time information at which the input data was input?

Consequences of failing to pay attention to the notes

- Feeding back input data obtained during operation to re-training without excluding outliers and missing data may lead to a decrease in the accuracy of the trained model.

● Ensure the security of input data during operation <Development>

- For input data in operation, has the system implemented a mechanism to detect and eliminate any data that may lead to abnormal behavior or have malicious intent?

Consequences of failing to pay attention to the notes

- If the input data obtained during the operation includes outlier data or missing data that leads to abnormal behavior, the probabilistic behavior of the AI may result in unexpected output data from the trained model. Not only will the accuracy of the learned model be degraded, but also the functionality required of the AI system cannot be ensured, which may lead to a decrease in the reliability of the AI system.

● Monitoring the safety of input data to feed back to relearning <Operation>

- In order to improve the reliability of AI systems, is it possible to monitor the input data fed back for relearning during operation for the presence of malicious data that may lead to performance degradation? Or, is it possible to eliminate malicious data before re-learning?
- If the malicious data cannot be eliminated before learning, can the system return to the state before the malicious data is learned?

Consequences of failing to pay attention to the notes

- If we fail to detect the inclusion of malicious data in the input data that feeds back to the relearning, the relearning model may be generated without reflecting the actual input data, and we may not notice it until the evaluation stage of the relearning model.

● Monitoring the safety of input data during operation <Operation>

- With regard to the input data used for inference during operation, is it possible to detect and eliminate any data that could lead to abnormal behavior or that is malicious?
- If malicious data cannot be eliminated before training, can the malicious data be reverted back to before training?

Consequences of failing to pay attention to the notes

- If we fail to detect the presence of malicious data in the input data used for inference during operation, we may not be able to notice the output of erroneous inference results. As a result, the accuracy of the retrained model will decrease. Furthermore, the degradation of accuracy may result in AI system stoppage, the production of defective products or missed products due to failure to detect abnormalities in AI system operation, or the possibility of danger to humans due to abnormal equipment operation.

7.6.26 Explanation of SQ-4

● Assessment of the adequacy of the safety operation of AI systems <Development/Operation>

- Is it possible to demonstrate the safety of the AI system by using statistical methods such as mean, variance, correlation coefficient, etc. to evaluate whether the safety operation mechanism can control the occurrence of abnormal behavior (abnormal output, data input leading to abnormal behavior, input of malicious data)?

Consequences of failing to pay attention to the notes

- Since operators cannot quantitatively evaluate and present the safety against abnormal operations, they may not be able to take priority measures against high-risk abnormal operations that occur frequently or have a large impact when they occur. As a result, the AI system may stop, defective products may be manufactured or missed due to the inability to detect abnormalities in AI system operation, or people may be endangered due to abnormal equipment operation.

7.6.27 Explanation of SQ-5

● Explainability of AI systems <PoC/Development/Operation>
<PoC>

- Did you use a simple model that is easy to explain as the algorithm used when developing the AI component?
- Did you use a simple model that is easy to explain as an algorithm to use when developing AI components, or did you use a simple model to approximate a complex model?
- Do you have a means of explaining the characteristics of the external environment and input data as a basis for the output data of the AI system? Or, is it possible to show the validity of the output data of the AI system using statistical methods?

<Development>

- Did you use a simple model that is easy to explain as an algorithm used in the development of AI components? Or, did you use a simple model to approximate a complex model?

- Does the AI system have a means of explaining the characteristics of the external environment and AI system input data as a basis for its output data?
- Does the AI system have a way to explain the characteristics of the external environment and AI system input data as a rationale for its output data, or is it possible to demonstrate the validity of the AI system's output data using statistical methods?

<Operation>

- Did you use a simple model that is easy to explain as an algorithm to be used when developing AI components?
- Did you use a simple model that is easy to explain as an algorithm used when developing AI components, or did you use a simple model to approximate a complex model?
- Does the AI system have a means of explaining the characteristics of the external environment and AI system input data as a basis for its output data? Or, is it possible to demonstrate the validity of the AI system output data using statistical methods, etc.?

Consequences of failing to pay attention to the notes

- When AI system output data is determined to be inappropriate, it is difficult to analyze the cause of the inappropriateness, which may prevent AI system developers and operators from making prompt corrections.

● Convincing AI Systems <PoC>

- Does the output data as an AI system match the customer's expected value? Or, even if it does not match the expected value, has the customer been convinced by showing the rationale for obtaining the output data as an AI system?

Consequences of failing to pay attention to the notes

- Clients may not have confidence in the AI system output data.

● Ease of understanding the AI system <Operation>

- Does the AI system output data match the customer's expected value? Or, even if it does not match the expected value, has the customer been convinced by presenting the rationale for obtaining the AI system output data?

Consequences of failing to pay attention to the notes

- Clients may not have confidence in the AI system output data.

7.6.28 Explanation of SQ-6

● Relevance of system behavior during operation <Development/Operation>
<Development>

- Have performance goals been defined for the AI system to determine that it is operating normally, such as the accuracy of the AI system output data?
- Is there a means to check the accuracy of the AI system's output data?
- Is there a means to check for changes in the characteristics of the AI system input data?
- Has the system determined an appropriate frequency for checking for degradation of the accuracy

of the AI system output data and changes in the characteristics of the AI system input data?
- Are methods for maintaining normal system operation in response to changes in the characteristics of AI system input data envisaged, for example, to prevent quality incidents from occurring?

<Operation>

- Are checks for changes in AI system input data characteristics and degradation of AI system accuracy performed at a defined frequency?

Consequences of failing to pay attention to the notes

- If performance goals for the AI system are not defined, it may not be possible to determine normal/abnormal conditions, which may lead to a decrease in accuracy of the output data as an AI system.
- If there is no means to check for changes in AI system input data characteristics or accuracy degradation, or if the appropriate frequency of checks is not specified, the AI system may not be able to respond quickly to the occurrence of accuracy degradation. As a result, AI system stoppage or failure to detect abnormalities in the operation of the AI system may result in the production of defective products or the occurrence of missed products.

7.6.29 Explanation of SQ-7

● Assumption of the amount of input/output data to be collected during operation <PoC/Development>

- During operation, is there a means to monitor the amount of input/output data to be stored as appropriate and to expand the amount of input/output data that can be stored?
- When input data and AI system output data labeled with correct/incorrect answers are continuously stored as a training data set for relearning, are there rules for storing and disposing of input/output data in advance to ensure that the amount of input/output data stored is appropriate and that the AI system can continue to learn properly?
- Do you have a rule for storing and disposing of input/output data in advance?
- In order to accumulate input data for re-learning, is the frequency of monitoring the amount of input/output data to be stored determined?

Consequences of failing to pay attention to the notes

- Input/output data collected during operation may no longer be stored, or unplanned storage expansion work may occur.

● Monitoring and controlling the amount of I/O data during operation<Operation>

- Is the system monitored at a specified frequency to ensure that the estimated maximum amount of stored input/output data specified in the system requirements is not exceeded?

Consequences of failing to pay attention to the notes

- Input/output data collected during operation may no longer be stored, or unplanned storage expansion work may occur.

7.6.30 Explanation of SQ-8

● Examination of various constraints on software and hardware <PoC>

- When using a neural network, show the relationship between the size of the network and the memory requirements to ensure that it can be operated on the hardware on which it will be implemented. (*The larger (more complex) the network, the higher the rate of correct answers, but the more memory is required.
- Examine whether quantization is necessary or not, or whether the weight information should be sparse or not. In doing so, we show the trade-offs between changes in calculation accuracy, memory, and degradation of the correct answer rate, and examine to what extent the performance degradation caused by quantization/sparsification is acceptable. (*Reducing the calculation precision reduces the memory used, but degrades the correct answer rate.
- We consider whether the programming language to be implemented should be the same as the one used in training.
- In the case of FPGA implementation, consider the trade-off between development time and performance in terms of the design method: manual design or high-level synthesis.
- If it is necessary to update the trained model during operation, we agree on the update method with the customer. When agreeing on the update method, we should also agree on who will perform the retraining and how the retrained model will be distributed.
  (e.g.) manually (ROM burning, hand delivery of memory card), automatically (network delivery), etc.

Consequences of failing to pay attention to the notes

- If the correct constraint conditions cannot be identified, the prescribed performance cannot be achieved during operation, which may result in rework of hardware selection and system configuration design.

● Formulation of hardware for operational use <Development>

- Has hardware been selected that meets system requirements such as the load and data volume when running AI models during operation?
- Has the hardware been selected so that it can be expanded when the input data increases in the future?

Consequences of failing to pay attention to the notes

- Without planning for the operational load and data volume, it may not be possible to select the appropriate hardware. As a result, there is a possibility of selecting over-spec hardware and architecture.

● Selecting software with updates in mind <Development>

- Do you consider the update frequency and support period of various software such as OS and OSS before using them?
- Do you decide how to respond to software updates and what to do when support ends?

Consequences of failing to pay attention to the notes

- If the update cycle of the OS or OSS used in the AI system is short, or if a problem is discovered after the external software used is no longer supported, the number of man-hours required to

respond to the problem on the AI system side during development will increase.

● Consideration of hardware constraints that depend on the performance and structure of the model <Development>

- The relationship between the size of the neural network and the memory requirement.
  - * The larger (more complex) the network, the higher the rate of correct answers, but the more memory required.
- * Indicates the trade-off between quantization (change in computational precision)/memory/correct answer rate degradation.
- * Indicates that reducing the calculation precision reduces the memory used, but degrades the performance of the correct answer rate.

Consequences of failing to pay attention to the notes

- If the correct constraint conditions cannot be identified, the prescribed performance cannot be achieved during operation, which may result in rework of hardware selection and system configuration design.

● Hardware maintenance based on the plan <Operation>

- Is the monitoring of compliance with the maximum amount of data to be stored and the load specified in the system requirements performed at a specified frequency?

Consequences of failing to pay attention to the notes

- Data may not be stored during operation.
- If hardware maintenance is not performed, AI system turnaround time may degrade, or unplanned AI system outages or storage expansion tasks may occur.

● Software updates <Operation>

- Is the system updated for various software updates such as OS and OSS, especially when there are security updates? Since the update cycle of AI-related OSS is short, it is important to consider the difference from the system development cycle.

Consequences of failing to pay attention to the notes

- If the software has not been updated, or if the system uses external software that is no longer supported and cannot be updated, it is likely that the problem will not be addressed by the developer of the external software even if it is discovered. Therefore, it may be necessary to take action on the AI system in operation. Furthermore, if the AI system cannot be addressed, it may have to be terminated.

● Consensus on the operational design of various software and hardware constraints <Operation>

- The operational design as a system on how to relearn and how to deliver the relearning is to be agreed with the customer.

Consequences of failing to pay attention to the notes

- If there is no operational design for the method of relearning and the method of delivering relearning, it is not possible to take prompt action when a decrease in accuracy occurs during operation. If the number of incorrect answers increases as the accuracy decreases, human checks

and corrections will be required, and the effectiveness of using the AI system will decrease.

7.6.31 Process Agility

In industrial systems, development was generally based on clear achievement goals and feasibility. Machine learning techniques, on the other hand, have uncertainty in both achievement goals and feasibility, so AI components and AI component development require iterative development and quality assurance.

Table 7.19 Considerations for Process Agility

ID	Point of view	Considerations	PoC	Dev.	Ops.
PA-1	In the development of AI components, is iterative development performed in sufficiently short iterative units, and is the cycle of quality improvement of AI models and AI systems sufficiently short?	● Ability to execute agile software development ・Are there any inadequacies in the internal environment and development procedures, such as customer cooperation, human resources, and equipment, that are necessary to carry out iterative software development like agile?	○	○	

Table 7.20　Considerations for Process Agility (cont'd)

ID	Point of view	Considerations	PoC	Dev.	Ops.
PA-2	Is continuous feedback on the operation status frequent	● Degree of cooperation of users of the system during the PoC of AI system ・Enough cooperation is obtained from users of the system so that the product owner can make appropriate decisions during the trial in PoC or from the trial results. ・Is it possible to get enough cooperation from the users of the system and to get continuous feedback from them so that the product owner can make an appropriate decision?	○		
		● Appropriate reflection of feedback from system users during AI system development ・Are the results of PoC trials, for which sufficient cooperation from system users was obtained, appropriately managed during development so that feedback can be provided?		○	
		● Appropriate reflection of feedback from system users during AI system operation ・Is the system able to confirm whether feedback from system users is being provided?			○
PA-3	Can releases and rollbacks be performed easily and quickly? When a problem occurs, is there a system to record and obtain information on the situation for cause analysis? Also, is it possible to reproduce the event based on the situation?	● Adequacy of AI system configuration management ・Is configuration management appropriately performed for versions of AI programs, training datasets, etc.?	○	○	○
		● Clarification and preparation of response measures, such as recording and acquisition of information and investigation of reproduction when problems occur ・Is the information to be recorded and the procedure/tool to be obtained for investigation when a problem occurs clear? Is there sufficient preparation of the environment for reproducing the event?	○	○	○
		● Release plan adequacy ・Is the release plan (e.g., canary release) of AI programs appropriately determined according to the characteristics of AI products and customer requirements?	○	○	
		● Quickness of Rollback ・Is there a mechanism to roll back promptly when an abnormality occurs in the released AI program?	○	○	○
		● Release according to the release plan ・Are AI programs released according to the release plan decided at the time of development? Also, is the release plan reviewed as necessary?			○

Table 7.21　Considerations for Process Agility (cont'd)

ID	Point of view	Considerations	PoC	Dev.	Ops.
PA-4	Is it expected to get better ・Is it possible to add new features quickly ・Is it possible to improve the model quickly ・Does it have a means to debug training and inference	● Is it expected to improve performance by gradual release? ・Is the performance expected to be improved by reviewing the data used for training, adding features, improving the model, etc.?	○	○	
PA-5	Is the development team equipped with human resources with appropriate capabilities ・Does it include "experts" in machine learning or data science, or domain experts?	● Ability to execute AI system development ・Are the human resources required for AI system development gathered and are the roles in the development system clarified?	○	○	
		● Development team in collaboration with the customer ・Does the customer side have people who understand business processes and domain knowledge, and are they incorporated into the development structure?	○	○	
PA-6	Is it possible to reflect experience in technology	● Reflection of experience from previous development ・Is a process and system established to reflect experience from existing AI application sites as technology in the next development?	○	○	○
PA-7	Are stakeholders outside the development team sufficiently convinced	● Are the system and roles of stakeholders clearly recognized ・Stakeholders outside the development team include management, different departments, and customers. Are the system and roles of stakeholders clearly recognized and communicated?	○	○	
		● Do you have an agreement or cooperative relationship with management and other departments?	○	○	
		● Is the field sufficiently convinced? ・Are the service users sufficiently convinced about the AI operation, and are they able to provide feedback?			○

Table 7.22 Considerations for Process Agility (cont'd)

ID	Point of view	Considerations	PoC	Dev.	Ops.
PA-8	Is there an update plan that considers the system lifecycle	● Relevance of AI system update plan ・Is an update plan developed that considers the entire lifecycle of the AI system, and is the update plan reasonable? The update plan should include OSS updates, responses to changes in input data characteristics, and responses to increases in input/output data volume.		○	
		● Are updates being made in accordance with the plan ? ・Does the company obtain feedback from customers on a case-by-case basis?			○

7.6.32 Explanation of PA-1

● Ability to perform agile software development <PoC/Development>

- Customer Cooperation: Is the product owner able to dedicate himself/herself to the project, is this understood by the organization to which the product owner belongs, and is he/she delegated the authority to make immediate decisions?
- Human resources: Do they have experience in agile development? Are there any issues with mindset or behavior?

It is difficult to predict the quality (especially performance indicators such as accuracy) that can be ensured in advance, so we have to explore how much quality can be achieved through a testing and exploratory process while repeating experiments. In some cases, it is necessary to adjust the requirements and use cases, and make a decision such as "accept the product if the reproduction rate is high even if the conformity rate is low.

Therefore, it is important to have stakeholders, including customers, understand the experimental and exploratory processes in advance and to obtain their cooperation. In this case, the following points should be taken into consideration for the customer's cooperation.

1. Can the product owner be dedicated to the project?
2. Does the organization to which the Product Owner belongs understand about 1.
3. Has the Product Owner been delegated the authority to make quick decisions?

Consequences of failing to pay attention to the notes

The final product will be different from what the customer wants to do, and the rework will be large.

7.6.33 Explanation of PA-2

● Cooperation level of system users during PoC of AI system <PoC>

It is necessary to agree in advance with the product owner on what to measure with regard to the output results of the system, data such as log information, and evaluation results from the users of the system, and to be prepared to make appropriate judgments. On top of that, the product owner should be able to make the necessary decisions periodically during the PoC and after the PoC is completed.

In order to facilitate the decision, it is recommended to agree on the indicator values in advance.

Periodically, for example, if significant data can be accumulated in a week, it is good to set up a place where a decision can be made once a week.

Consequences of failing to pay attention to the notes

The final product will be different from what the users of the system want to do, resulting in a lot of rework.

● Appropriate reflection of feedback from system users during AI system development <Development>

Before the start of development, the system developer should determine the priority in the development of the AI system based on the data collected in the PoC and agree with the product owner. It is also recommended to agree on the index values in advance in order to judge whether the output of the developed AI system is the expected result or not.

The development team members should proceed with the development according to the priorities agreed upon in advance.

The system developer should check the degree of achievement of the previously agreed indicator values and judge the appropriateness of the development.

Consequences of failing to pay attention to the notes

The end result will be something different from what the users of the system want to do, and the rework will be significant.

● Appropriate reflection of feedback from system users during AI system development <Operation>

It is necessary to agree in advance with the product owner on what to measure regarding the output results of the system, data such as log information, and evaluation results from the users of the system, and to be prepared to make appropriate decisions. On top of that, the product owner should be able to make the necessary decisions periodically during operation.

In order to facilitate the judgment, it is recommended to agree on the indicator values in advance. Periodically, for example, if significant data can be accumulated in a week, it is good to set up a place where a decision can be made once a week.

Consequences of failing to pay attention to the notes

Without a mechanism to provide appropriate feedback on incidents that occur during operations, continuous improvement cannot be achieved.

7.6.34 Explanation of PA-3

● AI System Configuration Management Adequacy <PoC/Development/Operation>

Are the hyper-parameter values (e.g., seed values of random numbers) and training datasets used for preprocessing and training recorded and managed historically so that the same AI model can be developed in order to enable anyone to isolate problems when they occur?

Consequences of failing to pay attention to the notes

Failure to obtain fixed results under the same conditions (lack of reproducibility) may lead to variable results in AI model accuracy comparisons. Time consuming trial-and-error,and errors will increase development time and lead to release delays.

Subparagraph ● Clarification and preparation of response measures, including recording and acquisition of information and investigation of reproduction in the event of a problem <PoC/Development/Operation>

The model is not updated in response to changes in data trends and accuracy deteriorates, or anomalies in prediction decisions occur due to data noise or model tuning errors. When these problems occur, it is necessary to have a system that can record the occurrence status and obtain it without omission in order to analyze the cause. For this purpose, it is recommended to clarify the information to be recorded and the procedures/tools to be used to obtain the information. In addition, in order to determine the validity of the cause of the analysis and to confirm the correction of the event, it is recommended to prepare an environment to reproduce the event in advance. Examples of information to be recorded: version of the model, parameters, contents of preprocessing/postprocessing, version of the data set

Consequences of failing to pay attention to the notes

Not only does it lead to incorrectly fixed problems and reduced accuracy/performance, but it also increases the time to investigate problems and delays the release schedule.

● Relevance of Release Plan <PoC/Development>

Are the hyper-parameter values used for preprocessing and training (e.g., seed values of random numbers) and the training dataset recorded and maintained for history so that the same training model can be developed so that anyone can isolate the problem when trouble occurs?

Consequences of failing to pay attention to the notes

(especially for Canary releases) If a new AI model, when run in production, is less accurate in its inference than the previous one or causes erroneous behavior, the entire user base may not be able to use the service until the rollback is complete.

● Fast Rollback <PoC/Development/Operation>

Is it possible to easily detect performance degradation in the broad sense of customer requirements when the performance is degraded by the added new category of training data sets? Also, is there a mechanism to automatically roll back to the previous version based on the detection results, or is there a mechanism to easily roll back using some kind of switch such as an environment variable?

The scope of configuration management for AI systems and AI components should be considered depending on their characteristics.

Consequences of failing to pay attention to the notes

Rollback is time-consuming or rollback failure may result in extended downtime, depending on the characteristics of the AI system.

● Release as planned <Operation>

Is the AI program released in accordance with the release plan determined during development? Also, is the release plan reviewed as necessary?

Consequences of failing to pay attention to the notes

If the AI program is not released as planned, it could have an overall impact on the development of the AI system, including rework and delays.

7.6.35 Explanation of PA-4

● Relevance of Release Plan <PoC/Development>

Although it is difficult to anticipate performance and performance improvement in advance, it is necessary to have a process and system in place that facilitates the review of data, addition of feature quantities, and model improvement in order to actually expect performance improvement as releases are made. In addition, in order to avoid endless pursuit of performance improvement, it is good to agree on a performance target that is commensurate with the cost in advance.

(e.g., to show that the performance improvement was achieved as expected. For example, in the PoC phase, in the planning

- It is necessary to determine the performance (timing, cost) as a condition of verification completion.
- If there are no conditions for completion of verification, there is a possibility that only the cost will be accumulated because there is no prospect of completing verification due to the pursuit of performance. (You will fall into PoC poverty.)

Consequences of failing to pay attention to the notes

The difficulty in reviewing data, adding features, and improving the model makes it time consuming and costly to improve performance. As a result, it may not be possible to achieve the performance goals commensurate with the costs agreed upon in advance.

Explanation of PA-5

● Performance of AI Product Development <PoC/Development>

The following skills are required.

- Doubt the evaluation metrics (evaluation results are overestimated) *Check if the current evaluation metrics are appropriate to achieve the objectives set in the requirement definition document.
- Doubt the training data set (e.g., the performance is not good enough for real problems). Check by cross-validation.
- Question the problem setting (e.g., the problem setting is inadequate and the goal cannot be achieved).
- Question the problem setting (problem setting is inadequate and the goal cannot be achieved).

Consequences of failing to pay attention to the notes

Unnecessary features are added to the system, which increases the engineering cost and computational resource cost.

Features that should not be added to the system are mixed in, obsolete features remain in the system, or data sources are lost, which may lead to operational failures such as unintended behavior.

● Development Team in Collaboration with Customers <PoC/Development>

In many cases, the team is divided into a team that develops machine learning models (which must also understand the customer's business processes, etc.) and a data scientist team for training data sets.

Even in a system divided into teams, it is good to clarify who is responsible for problems such as performance degradation. In addition, if the development team lacks knowledge of business processes or domains, it is desirable to enhance the collaboration between the development team and the customer by including experts from the customer side in the team.

Consequences of failing to pay attention to the notes

Unnecessary features are added to the system, which increases the engineering cost and computational resource cost. Features that should not be added to the system are mixed in, obsolete features remain in the system, and data sources are lost, which may lead to operational failures such as unintended behavior.

Explanation of PA-6

● Reflecting the experience of previous development/reflecting the knowledge of development <PoC/development/Operation>

- The experimental purpose, experimental conditions/environment, experimental results, and discussion should be clearly described.
- There should be a mechanism that allows anyone to reproduce the same training environment and accuracy (e.g., the training environment and hyperparameters (number of training sessions, learning rate, batch size, etc.) are described in a Dockerfile, etc.).

Consequences of failing to pay attention to the notes

In the development of AI systems, trial-and-error such as hyperparameter design requires more time than in conventional development. At that time, there is a possibility to shorten the time of trial and error by using the previous development experience. (= Without this, development cannot be carried out efficiently)

Explanation of PA-7

● Do you have a clear understanding of the stakeholder structure and roles ?<PoC/Development>
System/Cooperation: Do you have rules for escalation of opinions and inquiries so that opinions from customers/system users, such as "Things are different from usual, and there are more results that don't match our senses", can be quickly absorbed and improved?

Consequences of failing to pay attention to the notes

Lack of coordination results in unintended deliverable and interfaces, resulting in backtracking.

● Do you have an agreement or cooperative relationship with management and other departments? <PoC/Development>

System/Cooperative Relationships: Are the rules for escalation of opinions and inquiries established so that opinions from customers/system users, such as "Things are different from usual, and there are more results that don't fit our sense", can be quickly absorbed and improved?

Consequences of failing to pay attention to the notes

Lack of coordination, resulting in unintended artifacts and interfaces, and consequent backtracking.

● Do you have a clear understanding of the stakeholder structure and roles ?<Operation>

System/Cooperation: Are the rules for escalation of opinions and inquiries established so that opinions from customers/system users, such as "Things are different from usual, and there are more results that do not match our senses", can be quickly absorbed and improved?

Consequences of failing to pay attention to the notes

Failure to collaborate, resulting in unintended deliverable and interfaces, and consequent backtracking
.

Explanation of PA-8

● Reasonableness of the plan for updating the AI system <Development>

Is it planned at what timing the AI model/AI component should be updated? (Update by triggers such as when a failure occurs, when performance degrades, etc. OR Update only AI models or other parts as well)

An update plan that matches the development speed is necessary.

Consequences of failing to pay attention to the notes

In a system handled by industrial process data, it is highly possible that AI models/AI components and other components cannot be updated as easily as web services. Therefore, it is necessary to make a plan to update AI models/AI components together with updating other components of the system. If you continue to operate the system without updating it, you will be using it in a state where performance degradation and other problems have occurred, which may not meet customer expectations.

● Is it being updated according to the plan? <Operation>

Do you have a plan for when to update AI models/components? (Triggered by failure, performance degradation, etc., or periodically, etc.; update AI models only or other parts as well)

An update plan that matches the development speed is necessary.

Consequences of failing to pay attention to the notes

In a system handled by industrial process data, it is highly possible that AI models/AI components and other components cannot be updated as easily as web services. Therefore, it is necessary to make a plan to update AI models/AI components together with updating other components of the system. If the system continues to operate without updating, it will be used in a state where performance degradation and other problems have occurred, which may not meet the customer's expectations.

● Is the field sufficiently convinced? <Operation>

In order to improve the system, it is necessary to listen to the opinions of service users who actually use the AI system and improve it. It is necessary to build a system that provides feedback on the opinions of users.

Consequences of failing to pay attention to the notes

It may not be possible to make improvements from the perspective of service users without knowing whether the system is really helping to improve operational efficiency.

7.7 Quality Assurance Perspectives in AI product Development Process

In this chapter, we show the correspondence between the development process assumed in the IXI model (Figure 7.8) and the quality assurance perspective.

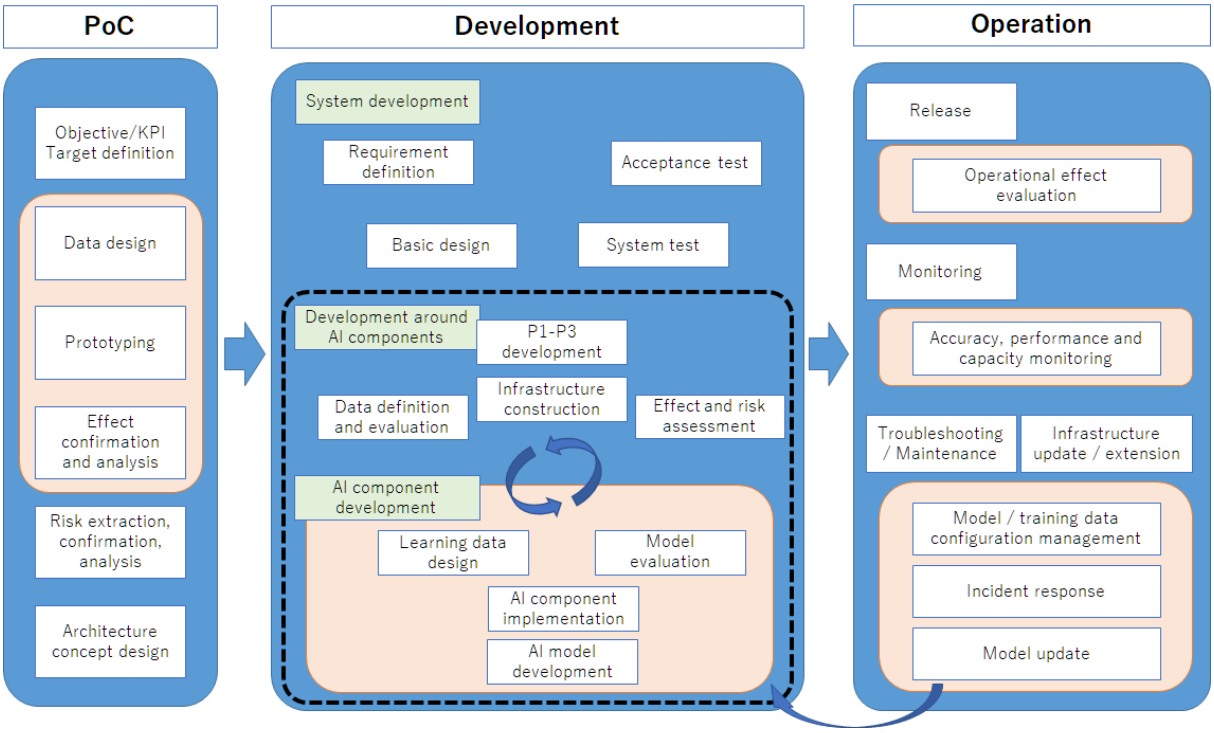


Fig. 7.8 Restated: Overall Process: IXI: Intelligent eXperimental Integration Model

7.7.1 PoC

PoC is an activity to demonstrate the feasibility of a new concept or idea. We will demonstrate what AI components and architecture can be used to meet the objectives and KPI targets.

The main activities of the PoC process in the IXI model are shown in Table 7.23.

The perspectives and their correspondence to the five axes in this process are shown in Table 7.24. The main assurance perspectives in the table are those that should be given special attention, and the related perspectives are those that should be referred to as necessary. In addition, those written with an asterisk, such as SQ-*, indicate that all perspectives of the axis are covered.

7.7.2 development process

In the development process, the system is designed and implemented as a real system based on the scope demonstrated by the PoC. The iteration cycle between the development process and the operation process may be very short depending on the environment dependency and the operation form of the AI component.

Table 7.23　Key activities in the PoC process

Activity	Overview
1. Definition of Objectives and KPI Targets	This is a process to set the objectives in PoC, not only for industrial systems, and to agree on them with the parties concerned. In this activity, the completion criteria for the development and verification of the verification target in the PoC are set. For industrial systems, the targets are set not only based on the KPI targets in terms of performance based on the operation results of the existing system, but also based on the business requirements such as the project cost associated with its feasibility and the acceptable risk of the system.
2. Data Design	This is an activity for target identification and collection of input data and training data sets to be used in AI component development for the tasks set by 1. In industrial systems, there are cases where data is required for a long period of time (e.g., degradation estimation) or where data with insufficient accuracy assurance is used. In this case, the scope and certainty to be confirmed during the PoC period should be planned, and a data collection and evaluation mechanism and review system should be prepared.
3. prototyping	The development of the AI component model, P1~P3, will be carried out based on the design in 2. It is extremely rare to introduce a module with uncertainty in an industrial system without verification, and the existing system part (P1~P3) may only collect data to control the risk of system quality degradation.
4. effectiveness check and analysis	1~3. based evaluation of the AI component. This evaluation will be based on measurement and evaluation conditions that take into account that industrial systems are subject to variations of 5M+E. Depending on the results of the analysis, the activities in 2~3. will be repeated.
5. risk extraction, confirmation, and analysis	Extract, confirm, and analyze risk targets based on the requirements and configuration of the target system, certification, and standards (e.g., functional safety standards, industrial control system security, etc.).
6. Design the overall architecture	concept for the development based on the requirements clarified by the activities such as data design, prototyping, effect confirmation, and analysis. This activity identifies the data, model, and configuration, and is used to align stakeholder rights and contracts.

To deal with the uncertainty of AI components, the development process is divided into three types: "system development," "AI component peripheral development," and "AI component development" (Table 7.25). These activities are iterative in nature, and quality assurance activities will be implemented to address the three characteristics.

In system development, the design and evaluation of P1~P3 and AI components are carried out based on the architectural concept. In conventional development, the activities corresponding to detailed design, implementation, and unit coupling test are equivalent to "AI component peripheral development" and "AI component development". Table 7.26 shows the main activities, and Table 7.27 shows the assurance perspectives.

In the development of the AI component periphery, the main activities are shown in Table 7.28 and the assurance perspective is shown in Table 7.29, where the P1-P3 and data infrastructure required for the AI component are developed and evaluated.

In the AI component development, activities such as algorithm implementation and model development are conducted to realize the behavior of AI components. Table 7.30 shows the main activities, and Table 7.31 shows the assurance perspectives.

SubsectionOperation Process

In the operation process, in addition to the conventional system operation, activities such as system

Table 7.24 Key assurance aspects in PoC

Key Activities	Considerations in Assurance	Key Assurance Perspectives	Related Assurance Perspectives
Objectives, KPIs, and Target Definition	Specifics of Objectives (Business Issues)	CE-1	SQ-1, MR-1, CE-8
	Clarity of Expected Effect, Demonstration Details	CE-6	-
Data Design	Mapping of Objectives to Data	CE-4	DI-2
	Ensuring Quality and Quantity of Data	DI-1,2	-
Prototyping	System Feasibility	MR-1	DI-1,2, MR-2,3,4,5,6,7, SQ-2,3, CE-3, PA-1,4
Effect Confirmation and Analysis	Certainty of Expected Effect	CE-1	SQ-1
	Assessment of Environmental Dependence	MR-1	DI-2, MR-2,5
Risk Extraction, Confirmation and Analysis	Acceptability and Risk Assessment for Stochastic Behavior	CE-2	CE-3,4, SQ-2,5
	Other Risk Assessment for System Quality	SQ-2	SQ-3,4
Architectural Conceptual Design	Requirements Based Configuration Identification	SQ-8	-
	Alignment of Rights, Inventiveness, etc.	CE-5,7	-

Table 7.25 3 types of development within the development process

Item	Overview
System Development	Clarify the achievement target including data collection by defining the requirement, and develop and evaluate the basic design, system test, and acceptance test in an iterative manner corresponding to the environment dependency.
AI Component Peripheral Development	Define, collect, and manage data, and conduct accuracy and risk assessment of AI components. Since it is impossible to collect data covering all conditions, we repeat data definition and evaluation activities such as narrowing down the conditions based on the target event and sufficiency of the amount of data collected by statistical evaluation of the data to confirm the scope of assurance.
AI Component Development	Using the obtained data, we develop and evaluate the necessary settings for the AI component, such as application algorithms, learning models, and hyperparameters. Depending on the results of the evaluation, the definition of the data to be collected (collection cycle, feature values, etc.) is changed, which is related to the development of AI components.

release, monitoring, trouble-shooting, maintenance, infrastructure update, and expansion are carried out, paying attention to environment-dependent data variation and inductive behavior of AI components. Table 7.32 shows the outline of the activities, and Table 7.33 shows the assurance perspective.

Table 7.26　Key Activities in System Development

Item	Overview
Requirement Definition	Determine the target of operation accuracy and the range to be guaranteed in P1 to P3 based on the purpose and target of AI components. Also, define the requirements based on System Quality. (For example, data management to meet security requirements)
Basic Design	Various components are designed according to the requirement definition. The basic design may determine the data to be targeted, and depending on the environmental dependencies and ease of explanation required by the system, a large number of prototypes may be performed here.
System Test	The implemented AI component peripheral system and AI components are evaluated in operation or on the desktop to evaluate the performance and accuracy intended in the basic design, and to evaluate operational requirements such as data collection. If it is judged that there is insufficient support for environmental dependency based on accuracy evaluation and collected data, iterate with requirement definition and basic design.
Acceptance Test	Reevaluate to determine implementation in the field environment and confirm the operation and management method. For example, activities such as checking the behavior of the system under non-target environmental conditions and its response, data management, and confirmation procedures are conducted. When model updating is assumed in the operational environment, procedures such as preconditions for updating and evaluation conditions are evaluated in this activity.

Table 7.27　Assurance perspective in system development

Key Activities	Assurance Considerations	Key Assurance Perspectives	Related Assurance Perspectives
Requirements Definition	Defining the Use Environment and Use Cases	SQ-2	SQ-3,4,6,7,8
	Define System Configuration Conditions	SQ-8	SQ-6,7
	Risk Management	SQ-2	CE-2,5,7,8, SQ-4
	Setting Quality Objectives	CE-1	SQ-1,2,3,4, MR-1
Basic Design	Fail Safe Design of Systems	SQ-3	DI-4, SQ-4
	In-Process Evaluation Mechanism Design (Canary Release)	PA-1	PA-2,3
	Rollback Design for In-Production Incident Response	SQ-3	PA-3
	Operational Monitoring Design	SQ-6	DI-4, SQ-1,7, PA-2,3
	Design Components Based on Requirements	SQ-8	SQ-7
System Test	Evaluation Against System Requirements Quality	CE-1	SQ-*, PA-3
	Logging Data Sufficiency Evaluation for Explanatory and Environmental Dependencies	SQ-5	DI-1,2,4,5, SQ-4
	Fail-Safe Testing of Systems Requiring Behavioral Verification	SQ-3	SQ-2,4
	Assessing the Sufficiency of Operational Monitoring	SQ-6	SQ-7
Acceptance Test	Conformance Assessment of Operational Requirements (Management Procedures, etc.) in the Operational Destination Environment	PA-2	MR-2, SQ-*, PA-3,8
	Evaluation of Environmental Dependencies in the Operational Destination Environment	DI-2	MR-5, SQ-6,7,8

Table 7.28　Key activities in the development around AI components

Item	Overview
P1-P3 Development	Based on the basic design, develop P1~P3. Collaborate with infrastructure construction to collect data necessary for AI component development and evaluation.
Infrastructure Construction	Collect and manage data used for implementation and evaluation of AI components, and build an infrastructure environment for continuous data management.
Data Definition and Evaluation	Define training data to be used for AI models and evaluation data to be used for effectiveness and risk assessment. Then, we evaluate how well the data items, data sets, and features correspond to the environment dependency based on the actual data.
Effectiveness/Risk Evaluation	Based on the actual operation of the AI component, evaluate the requirement-based effectiveness/risk (e.g., prediction accuracy, execution performance) based on the collected data.

Table 7.29 Assurance perspective in the development around AI components

Key Activities	Assurance Considerations	Key Assurance Perspectives	Related Assurance Perspectives
P1-P3 Development	P1: Rule Implementation for Input Assurance	DI-2, SQ-3	DI-4,5
	P2: Run-time Monitoring, Redundancy Implementation	SQ-3	SQ-7
	P3: Implement Rules for Output Assurance	SQ-3	DI-5, SQ-6
Infrastructure Construction	Data Collection during Development and Operation, Mechanism for Model Evaluation	DI-1,2	CE-8, DI-3, MR-1,2,7
	Monitoring During Operation, Support for Rollback	SQ-6	DI-4,5, SQ-1,7, PA-2,3
	Data Privacy, Security	CE-5, SQ-3	DI-5, SQ-2,4
	Building Systems for Operational Scalability	SQ-7	PA-8
	Is there a mechanism for teaching people such as annotations and labels	DI-2	-
	Data Definition, Model and SW Configuration Management	PA-3	SQ-8
Data Definition and Evaluation	Feature Validation	DI-2	-
	Data Privacy, Security Check	CE-5	DI-5, SQ-2,3,4
	Validation of Evaluation Data (Test Data)	DI-2,3	MR-1,3
Effectiveness and Risk Assessment	Selection of Appropriate Methods for AI Component Behavior Validation (Metamorphic Testing, Statistical Evaluation)	SQ-6	DI-2, MR-4, SQ-1, PA-4
	Misjudgment, Confirmation of Behavior against Unexpected and Correspondence to Operational Methods	SQ-3	DI-4, SQ-2,6
	Assessment of System Safety	SQ-3	SQ-2,4
	Evaluation of Prediction Accuracy and Execution Performance	SQ-6	MR-1,2,5

Table 7.30 Key activity points in AI component implementation

Item	Overview
Training Data Design	Design the training data necessary to create the model used by the AI component. To make the data suitable for training, it is necessary to prepare the data by cleansing and padding. We design the mechanism of this data preparation process.
Implementation of AI Component	This class implements algorithms for AI component to work in real system. The implementation includes processing of input data to the algorithm and selection of hyperparameters so that the AI component can output the required index.
AI Model Development	Develops models to be used by the AI component. data collected in the development around the AI component is turned into data sets based on the training data design, and models are developed to enable the AI component to operate.
Model Evaluation	Evaluate whether the AI component and the AI model can cope with environment dependency and explainability in the condition that the AI component and the AI model can work together.

Table 7.31 Warranty perspective on AI component implementation

Key Activities	Assurance Considerations	Key Assurance Perspectives	Related Assurance Perspectives
Training Data Set Design	Training Data Set Validation	DI-*	CE-4, MR-5
	Correctness Check of Annotated and Labeled Data	DI-2	-
	Validity Check of Evaluation Data (Test Data)	DI-3	DI-5
	Are cleansing, padding, and data generation methods appropriate?	DI-1,2	-
AI Component Implementation	Validation of Hyperparameter Selection	MR-4	-
	Validation of Input Data Processing (Algorithm Specific)	DI-2	-
AI Model Development	Validity of Training Method	MR-4	MR-3
	Configuration Management of Training Datasets, Hyperparameters, etc.	MR-4, PA-3	-
Model Evaluation	Are we able to observe and confirm changes in the internal state from training to prediction? (DNN coverage, etc.)	MR-4	-
	Confirmation of Actual Values against Expected Prediction Accuracy	MR-1,2,3	-
	Evaluating Whether Environmental Dependence and Explainability Can Be Addressed	SQ-5	-

Table 7.32　Key activities in the operational process

Item	Overview
Release	After the developed system is installed at the site where it will be operated and its operation is confirmed, an evaluation based on the operation site data is conducted. The evaluation conditions shall be clarified as the evaluation conditions for the operation site based on the environment dependency and operation requirements.
Evaluation of the effectiveness of an operation	At the time of release, it is evaluated whether the effect assumed at the time of development can be obtained for the operational environment, and the explanatory nature of the decision to start operation is prepared.
Monitoring	Monitor the input/output behavior of the system and AI components to detect unexpected behavior due to factors such as environment dependency. Collects information necessary for response when an incident occurs.
Accuracy, Performance, and Capacity Monitoring	AI components can lose accuracy due to environmental dependencies. Accuracy degradation should be monitored by means such as comparison with P2 and variation in behavior over time. At this time, since a large amount of data may be required to ensure explanability, infrastructure performance and data capacity should also be monitored.
Troubleshooting / Maintenance	Investigate and determine responses to stochastic behaviors based on collected data. In conventional troubleshooting, it is common to reproduce the behavior in the development/verification environment, but it is not possible to reproduce the behavior depending on the data collection infrastructure and models for environmental dependency. It is necessary to clarify what level of reproducibility is necessary for troubleshooting and maintenance.
Infrastructure Update/Expansion	Respond to data collection requirements that may change under the operational environment. For example, update target data definitions and expand data capacity to accommodate long-term storage of data not envisioned at the time of development.
Model and Training Dataset Configuration Management	In situations where models are updated during operation, the models and training datasets used by AI components are stored correspondingly to prepare for ease of explanation.
Incident Response	In some cases, emergency rollback decisions are made in response to a problem and the model is updated. In this case, an explanation should be provided as to whether the rollback really improves the accuracy.
Model Update	Based on the evaluation of the impact of the model used by the AI component on the System Quality, the model used for the operation is updated. If there is a risk to the sufficiency of the model evaluation, for example, the AI component may be operated redundantly to evaluate the performance of the new model.

Table 7.33 Key activities in the operational process

Key Activities	Assurance Considerations	Key Assurance Perspectives	Related Assurance Perspectives
Release	Minimizing Damage in the Event of Post-Release Problems	PA-3	CE-7
Evaluate Operational Effectiveness	Release Interval to Allow Time to Adjust for and Evaluate Environmental Dependencies	PA-3, 8	PA-7, SQ-8
	Does the Operational Environment Provide the Benefits Assumed During Development	CE-1, SQ-1	CE-3
Monitoring	Monitor Inputs and Outputs to Check for Abnormalities	DI-5, MR-1,3,9, SQ-6	CE-4
Accuracy and Performance Capacity Monitoring	Store data and logs appropriately, taking into account capacity, privacy, and security	SQ-3	CE-4,5, DI-2, SQ-7
Troubleshooting and Maintenance	Reproducible in Development and Verification Environments	MR-2	-
Infrastructure Update/Expansion	Are the resources (GPU, HDD (data area), network (communication environment)) expandable as needed	SQ-7	-
Model and Training Dataset Configuration Management	Configuration Management with Combination of Model, Real Configuration and Data	SQ-7,8, PA-3	PA-7
Incident Response	Determine the need for investigation, correction, etc. based on probabilistic behavior	CE-2	CE-4, PA-7
	Explanations for Results and Frequencies that Differ from Customer Knowledge Inferences	CE-4, 6	CE-8, PA-7
Model Update	Does Target Performance Still Apply with Additional Data	DI-1	CE-3, DI-3, SQ-8, PA-6,7
	Can we handle the risk of data oblivion (model rollback)	PA-3	CE-3, SQ-8, PA-6,7
	Are the Features Reasonable	MR-3	CE-3, SQ-8, PA-6,7
	Is it reasonable to change training algorithm, hyperparameter, etc.	MR-4	CE-3, SQ-8, PA-6,7

7.8 Example of Quality Assurance Reviews

In this chapter, we show an example of quality assurance study based on a published case study. The paper "Development of AI Technology Mountable on Machine Controller" [5] by OMRON Corporation was selected as a case study with specificity available at the time of guideline formulation. In this paper, the conceptual verification of the AI system was conducted using a horizontal pillow wrapping machine as the subject.

Quality assurance needs to decide the goal and approach depending on the business and contract type. Since the above published case study is a conceptual verification and no such information is available, the following assumptions are made for the business and contract type.

1. Add AI component to packaging machine (equipment)
2. The customer is responsible for manufacturing and operating the equipment
3. AI product development is based on the premise of a phased contract and process in the IXI model

7.8.1 Quality assurance in PoC

From the technical information in the paper, we considered that there are quality assurance considerations shown in Table 7.34 to Table 7.38.

Table 7.34　Example of Quality Assurance Study with Objective, KPI and Target Definition

QA Considerations	ID	Description	Examples of Case Studies
Concreteness of Objectives (Business Issues)	CE-1	● Clarification of Business Issues to be Solved by AI ● Possibility of Solving Business Issues with AI ● Effectiveness of Solving Business Issues to be Solved by AI ● Clarification of PoC Completion Criteria ● Understanding of continuous improvement for maintaining AI performance	● The objective is to prevent "seal misalignment" that causes "film snaking" phenomenon during packaging. ● The purpose is to prevent "seal misalignment" that causes "film snaking" phenomenon during packaging. "Film snaking" should be detected before "seal misalignment" occurs to maintain the equipment and prevent defective products. ● By learning the normal state of the equipment, it is possible to detect abnormalities in the equipment in real time. ● It is expected to maintain the equipment and prevent the generation of defective products. ・As a risk, it is no problem to stop the equipment instantaneously when countermeasures are taken. It is more problematic if even one defective product is produced, so please strictly monitor for signs of "film meandering". ・criteria for PoC are data collection, algorithm verification, and confirmation of effectiveness.
Expected effect, clarity of demonstration	CE-6	● Level of understanding of AI model's explanatory difficulty ● Level of understanding of AI's explainability and accuracy ● Level of understanding of model selection	● Stochastic behavior was explained to customers, and they understood and agreement was obtained. ● At the completion of PoC, the training data set, algorithm, test results and analysis were explained. ● Under the condition that the proposed model and algorithm would not affect the control of the device, the proposed model and algorithm were explained and the agreement was obtained.
Evaluation Indicators of AI Model	MR-1	● Definition of Tentative Specification of Indicator Values ● Investigation of Generalization Performance	● The indicators of correct answer rate, overlook rate, and miss rate were explained to the customer and agreement was obtained. We also interviewed the customer about their requirements for the index values. ● The generalization mechanism of the AI model was explained and agreement was obtained.

Table 7.35　Example of quality assurance study in data design

QA Considerations	ID	Description	Examples of Case Studies
Mapping of Objectives to Data	CE-4	● Understanding of the need for data consistent with the customer's business challenges ● Understanding of the quality and quantity of data required for AI training ● Customer understanding of changes in input data trends during operations	● Problem Explained to customers and obtained their agreement on the data required for resolution (AI training, testing). Explained the necessity of acquiring multiple data that are causally related to "film meandering". ● The number and quality of data required for training was explained to the customer based on the characteristics of the AI algorithm, and agreement was obtained.
Securing data quality and quantity	DI-1,2	● Securing the amount of data required for AI training ● Securing data to be used for cross-validation, generalization performance, etc. ● Evaluation of "bulking up" data	● Details on the data to be handled were included in the PoC plan. [Quality [Quality and Quantity] 400 workpieces under normal conditions and 100 workpieces under abnormal conditions in actual operation data were considered to be sufficient. [Cross-verification] Cross-verification shall be conducted. [Bulking] "Bulking" is not performed.

Table 7.36　Example of quality assurance study in prototyping

QA Considerations	ID	Description	Examples of Case Studies
Feasibility of the System	CE-3	● Level of Understanding of Agile Software Development ・Explanation of System Architecture	● The agile development process was clearly explained to the customer.

Table 7.37　Example of quality assurance study in effect verification and analysis

QA Considerations	ID	Description	Examples of Case Studies
Certainty of Expected Effects	CE-1	● Clarification of Business Problem to be Solved by AI ● Possibility of Solving Business Problem by AI ● Confirmation and Analysis of Effectiveness of Solving Business Problem by AI	● Results of PoC Implementation The results of the PoC were summarized in a report and explained to the customer. Here, we reported the probability of detecting "film meandering" in the acquired data. → The demonstration and explanation were repeated until the customer's consent was obtained.
Evaluation to environmental dependency	MR1,2,3,6	● Validation of the learning process ● Validation of the structure of the AI model ● Recording of hyperparameters ● Identification of noise affecting the AI model ● Validation of the measurement method	● The results of the PoC were summarized in a report and explained to the customer. It was confirmed that the correct answer rate met the customer's requirements. The hyperparameters set by the engineers were explained to the customer, and the customer understood that the values were set based on the data characteristics. Noise factors such as environmental fluctuations were identified. We explained to the customer that the selection of sensing locations and sensors was appropriate at the time of PoC.

Table 7.38　Example of quality assurance study in risk extraction, confirmation and analysis

QA Considerations	ID	Description	Examples of Case Studies
Acceptability and Risk Assessment for Stochastic Behavior	CE-2,3,4 SQ-5	● Level of Understanding of AI Output from Stochastic Behavior ● Risk Tolerance of Results Output from Stochastic Behavior ● Clarification of Responsibility for AI Product Output Understanding of agile software development ● Understanding of the need for training data that matches the customer's business problem ● Understanding of the quality and quantity of training data required to train the AI model ● Customer understanding of changes in input data trends during operation ● Explainability of the AI system ● Convincibility of the AI system	● Explanation of the stochastic behavior to the customer, understanding and agreement obtained. ● Data required for problem solving (AI training, testing) was explained to the customer and consent was obtained. The agile development process was clearly stated and explained to the customer The quality and quantity of the training data were described in the PoC results report and reported to the client. ● Explained the AI algorithm employed and helped the client to understand the explainability and acceptability of the results calculated by the AI; this has been included in the PoC results report.
Other risk assessment for system quality	SQ-1,2,3,4	● Consideration of customer value for AI products ● Extraction of risks for AI systems	● The results of the PoC were summarized in a report and explained to the customer. In this report, we explained the effectiveness, safety, and maintainability of the system. ・Risks were identified using a risk management table, and the extent of damage and methods of dealing with it were shared with the client and agreed upon. ・It was agreed with the customer that the risk management chart should be updated sequentially. ・A review of the risks was conducted with the customer in the PoC report.

Table 7.39　Example of quality assurance study in architectural conceptual design

QA Considerations	ID	Description	Examples of Case Studies
Identification of configuration based on requirements	SQ-8	● Consideration of various constraints on software and hardware	● Hardware specifications at the time of PoC implementation were described and explained in the PoC plan. Hardware specifications during operation were also considered and described.
Alignment of rights, inventiveness, etc.	CE-5,7,8	● Contractual arrangements regarding the attribution of intellectual property rights and other rights and conditions of use contained in PoC deliverables ● Understand the ownership and terms of use of intellectual property rights included in PoC deliverables ● Clarification of the security level, scope of information disclosure and handling restrictions of customer data used for AI ● Agreements on the ownership of rights and conditions of use for data used in PoC ● Clarify responsibilities in the event of problems arising from PoC ● The degree of customer cooperation and involvement in the development of AI products	● The attribution of copyrights and other intellectual property rights included in the PoC deliverables and the conditions of use were explained to the customer and agreed upon in the basic contract or other agreement after obtaining their understanding. The terms and conditions of use of the control data, etc. required for PoC were stipulated in a ● non-disclosure agreement or basic agreement. Liability issues in the event of PoC problems were defined in the basic agreement. ● The role of the customer is clarified with respect to the provision of equipment control data and the scope of provision of customer equipment design information necessary to proceed with PoC.

7.8.2 Quality assurance in development

Depending on the requirements and architecture of the equipment to be developed, we considered the quality assurance considerations shown in Table 7.40 to Table 7.51.

Table 7.40 Example of quality assurance study in requirements definition

QA Considerations	ID	Description	Examples of Case Studies
Defining the Use Environment and Use Cases	SQ-*	SQ	● Explained to the customer what data is needed to solve the problem. Data such as film transfer main shaft torque, speed, position, ... are required ● Defined the data acquisition method and management method. Data is acquired from the controller via the network, and the data is managed on the server of the development company. ● The requirement specification was written up again for the following main points. Introduction effect, risk, safety, maintainability, explainability of the algorithm, handling during operation, etc. - The list of terms and their explanations were clearly stated and agreed in the specification.
Define System Configuration Conditions	SQ-8	● Formulate hardware for operational use ● Selection of OSS in consideration of updates ● Examination of hardware constraints dependent on model performance and structure	● Controller calculation speed, data acquisition frequency, communication environment, hardware configuration, etc. were clarified and described in the requirement specification. ● The handling of OSS was also described. ・The warranty range of P1 to P3 was defined and described in the requirement specification. P1: Data cleansing is performed to prevent outliers from entering AI component and P2. P2: Perform "film meandering" detection on input data by a means different from the AI component. P3: Determine the final output from the output results of the AI component and P2. Outlier outputs are removed. ● The CPU and memory constraints were determined so that the inference program would not affect the device control.
Risk Management	CE-5,7, SQ-2	● Contractual arrangements regarding intellectual property rights, etc. for AI products ● Understanding of intellectual property rights, etc. for AI products ● High security of customer data used in AI systems, scope of information disclosure and ● Security of customer data used in AI systems, scope of disclosure, and restrictions on handling ● Rights to data contained in AI products ● Clarification of customers and stakeholders ● Clarify responsibility for the development process and output of AI products ● Consideration of risk estimates for AI systems	● After explaining the ownership of intellectual property rights, including copyrights, contained in the AI product and the conditions of use to the customer and gaining their understanding, an agreement was reached through a basic contract, etc. ● The terms and conditions of use of control data, etc., required for development were stipulated in a basic contract, etc. ● The role of the customer, such as the provision of control data, was clarified. ● The basic contract defines the responsibility in case of problems in development and the responsibility for the content of the output of AI components. The risk management table was updated at this point.
Setting Quality Objectives	CE-1, SQ-*, PA-4	● Clarification of the business problem to be solved by AI ● Possibility of solving the business problem by AI ● Effectiveness of solving the business problem to be solved by AI ● Prospect of performance improvement through PoC and staged release	● Clarified and described again in the requirement specification. It was decided that signs of "film snaking" would not be missed by real-time detection. ● It was stated in the requirement specification that the result of PoC showed the prospect of solving the customer's problem by continuous improvement of the AI system.

Table 7.41　Example of quality assurance study in basic design

QA Considerations	ID	Description	Examples of Case Studies
System Fail-Safe Design	DI-4, SQ3,4	● Ensuring Safety for AI Products ● Ensuring Control Mechanisms to Prevent Abnormal Output ● Ensuring Maintainability (the ability to detect, diagnose, and repair failures and abnormalities) for AI products ● Ensure safety for data fed back to relearning ● Ensure safety of input data during operation	● Warn of signs of "film meandering" but do not affect control. ・It shall not affect the normal operation of the PLC. ● In the case of an abnormal output that cannot be judged as "film meandering," what is abnormal shall be presented and the operator's judgment shall be sought. ● Even if there is no AI product, it does not affect the normal operation. ● Training datasets should be verified for their data characteristics before training. ● If incorrect data is input, the safety function of the controller will stop the device.
Design of mechanism for in-process evaluation (Canary Release)	PA-1,3	● Ability to perform agile software development ● Adequacy of configuration management of AI products ● Adequacy of release plan ● Rapidity of rollback	The phased release schedule was described in the basic design document, explained, and agreed upon. Since the learning model is updated in accordance with the periodic data acquisition, the release is made in accordance with the timing. ☒ We also described the configuration management method so that the rollback can be implemented quickly.
Rollback design for in-service incident response	SQ-3, 4, 6, PA-3	● Ensuring safety of AI products ● Ensuring control mechanisms to prevent abnormal output ● Ensuring maintainability of AI products (ability to detect, diagnose, and repair failures and abnormalities) ● Maintainability of AI products (ability to detect, diagnose, and repair failures and anomalies) ● Security of data fed back to learning ● Validation of safe operation of AI products ● Validation of system operation after operation ● Validation of configuration management of AI products ● Validation of release plans ● Rapidity of rollback	● Since it is easy to change the algorithm inside the controller, it is confirmed that the design is rollbackable.
Operational Monitoring Design	SQ-1,6, PA-2	● Consideration of customer value for the AI system ● Relevance of system operation after operation ● Appropriate reflection of feedback from users of the system during AI system development	● During operation The system is designed so that the status of data acquisition and monitoring variables can be visualized in the controller. ● After operation, agreement is obtained from the customer to perform data acquisition and analysis in a timely manner.
Design of components based on requirements	SQ-8	● Formulation of hardware assuming operation ● Selection of OSS in consideration of updates ● Examination of hardware constraints depending on model performance and structure	・P1-P3, AI Components were designed and described in the basic design document. ● The amount of data during operation is collected in a timely manner from the data stored in the customer's data server. ● The calculation speed, data acquisition frequency, communication environment, and hardware configuration of the controller were clarified and described in the requirements specification. (Existing) ● Handling of OSS was also described. (Existing)

Table 7.42　Example of quality assurance study in system testing

QA Considerations	ID	Description	Examples of Case Studies
Evaluation for System Requirements Quality	CE-1, SQ-*, PA-1,2,3	● Clarification of business issues to be solved by AI ● Possibility of solving business issues by AI ● Effectiveness of solving business issues to be solved by AI ● Ability to perform agile software development ● Appropriate reflection of feedback from users of the system during AI system development ● Appropriateness of configuration management of AI products ● Appropriateness of release plans ● Speed of rollback	● Test contents to confirm that the requirements are met are described in the test specification. ● Tests were conducted based on the test specification, and it was confirmed that the required specifications were met.
Logging Data Sufficiency Assessment for Explanatory and Environmental Dependencies	DI-1,2,4,5, SQ-4,5	● Securing the amount of data required for AI training ● Securing data used for cross-validation, generalization performance, etc. ● Evaluating "bulky" data (L ● Evaluation of "bulky" data ● Alignment of data used for training with business problem ● Quality of training dataset ● Characterization of training data ● Complexity of data definition (assumed model) for business problem (phenomenon) ● Appropriateness of route and management ● Correctly assigned labels and correct values ● Confirmation of consistency of input data obtained from multiple subsystems ● Appropriateness of safe operation of AI products ● Explainability of AI products	In order to explain that the test results meet the required specifications, the acquired data and system output results were described in the test report and explained to the customer. ● Under what conditions and with what kind of data, the client understood what kind of results would be obtained. ● Output data was carefully examined (labeled, normal, abnormal) and explained to the customer. ● It was decided not to "bulge" the data. ● The AI algorithm was selected to select explanatory variables without multicollinearity. ● Before training, the distribution characteristics of the training data were checked, and preprocessing such as outlier removal was performed. ● We confirmed that the data collected from the device was collected without any omissions. ● The characteristics of the training data set were discussed with the customer, and labeling was performed. We also confirmed the consistency of the data, including the consistency of the units. ● Explained to the customer that the algorithm based on the decision tree made sense.
Fail-Safe Testing of Systems Requiring Behavioral Verification	SQ-2,3,4	● Considering Risk Prediction for AI Systems ● Ensuring Safety of AI Systems ● Ensuring Control Mechanisms to Prevent Abnormal Output ● Ensuring Maintainability of AI Products ● Maintainability of AI products (ability to detect, diagnose, and repair failures and abnormalities) ● Maintain safety of data fed back to relearning ● Maintain safety of input data during operation	● Perform fail-safe testing of basic design. Make sure that AI products do not affect the system this time.
Evaluation of the sufficiency of operational monitoring	SQ-6,7	● Validity of system operation after operation	● As a method of operational monitoring, data during operation shall be mapped and compared in a timely manner to confirm the effectiveness of the AI product. The customer's agreement on this was obtained.

Table 7.43　Example of Quality Assurance Study in Acceptance Testing

QA Considerations	ID	Description	Examples of Case Studies
Conformance Evaluation of Operational Requirements (Management Procedures, etc.) in the Destination Environment	CE-1, MR-2, SQ-*, PA-2,3,8	● Clarification of Business Issues to be Solved by AI ● Possibility of Solving Business Issues by AI ● Effectiveness of Solving Business Issues by AI ● Generalization performance goals ● Methods for measuring generalization performance SQ ● Customer cooperation in AI product development ● Relevance of AI product configuration management ● Relevance of release plan ● Rollback speed ● Adequacy of AI Product Update Plan	● Test contents to confirm that the required specifications are met even in the destination environment were described in the test specification. The key points are whether the issues have been resolved and whether the generalization performance is met. ● Conducted tests based on the test specifications to confirm that the required specifications are also met in the destination environment. ● Confirming that the established shipping judgment criteria are met. ・Configuration management, release plans, etc. are described in the requirement specification or maintenance contract and agreed upon by the customer.
Evaluation of environmental dependencies in the customer's environment	CE-4, DI-2, MR-5, SQ-7,8	● Understanding of the need for data that meets the customer's business challenges ● Understanding of the quality and quantity of data required for AI training ● Customer understanding of changes in input data trends during operation ● Consistency of data used for training with business challenges ● Ensuring quality of training datasets ● Evaluation of training data characteristics ● Complexity of data definition (assumed model) for business challenges (phenomena) ● Relevance of data acquisition routes and management Appropriateness of data acquisition route and management ● Correct labeling and correct answer values ● Appropriateness of AI noise tolerance (robustness) ● Assumption of data volume to be collected during operation ● Development of hardware for operation Select OSS in consideration of updates	● As an operational monitoring method, map and compare data during operation in a timely manner to confirm the effectiveness of AI products. ● Explained the acquired data and the output results of the system to the client in order to explain that the test results meet the required specifications. In particular, robustness was evaluated here.

Table 7.44　Example of quality assurance study in P1-P3 development

QA Considerations	ID	Description	Examples of Case Studies
P1: Rule Implementation for Input Guarantees	DI-1,2,4,5, SQ-7	● Securing data for cross-validation, generalization performance, etc. ● Consistency of data used for training with the business problem ● Ensuring quality of training data set ● Training ● Complexity of the data definition (model to be assumed) for the business problem (phenomenon) ● Adequacy of the data acquisition route and management ● Correctly labeled and correct values ● Feasibility of the mechanism for outliers ● Confirmation of consistency of data obtained from multiple subsystems ● Assumption of the amount of data to be collected during operation	● Behavior and mechanism of P1 are described in each component detailed design document. ・Algorithm to remove outliers for each data was described. ☒ Explained how this results in guaranteed input to the P2 and AI components.
P2: Implementation of In-Process Monitoring and Redundancy	SQ-2,3	● Consider risk prediction for AI products ● Ensure safety for AI products ● Ensure control mechanisms to prevent abnormal output ● Ensure maintainability (ability to detect, diagnose, and repair failures and abnormalities) for AI products ● Ensure safety of input data during operation ● Maintainability of AI products (ability to detect, diagnose, and repair failures and abnormalities) ● Security of data fed back to training ● Security of input data during operation	● P2 behavior and mechanism are described in each component detailed design document. ・Data processing other than AI component processing was described. ・It was described that it operates independently from AI component. Exception handling to eliminate invalid data is incorporated in the AI component.
P3: Implementing Rules for Output Assurance	SQ-2,3	● Considering Risk Prediction for AI Products ● Ensuring Safety of AI Products ● Ensuring Control Mechanisms to Prevent Abnormal Output ● Ensuring Maintainability of AI Products (Ability to Detect, Diagnose, and Repair Faults and Anomalies) ● Security of data to be fed back to relearning ● Security of input data during operation	● Behavior and mechanism of P3 are described in each component detailed design document. ・The process to determine the final output from the output result of AI component and P2 was described. The exception processing to eliminate invalid data was incorporated in the AI component.

Table 7.45　Example of quality assurance study in infrastructure construction

QA Considerations	ID	Description	Examples of Case Studies
Mechanisms for Data Collection and Model Evaluation during Development and Operation	CE-8, DI-1,2,3, MR-1,2	● Degree of Customer Cooperation and Involvement in AI Product Development ● Prepare Data for Cross-validation, Generalization Performance, etc. ● Evaluation of data argumentation ● Alignment of data used for training with business problems ● Ensuring quality of training data sets ● Evaluation of training data characteristics ● Complexity of data definitions (assumed models) for business problems (phenomena) ● Validation of data sources and management ● Labels and Answers' correctness ● Independence of data used for cross-validation, generalization performance, etc. ● Validity of training results ● Targets for generalization performance ● Methods for measuring performance	● Methods for obtaining and managing data during development are specified in the requirements specification and basic design document for development, and in the maintenance contract for operation, and the system was designed to collect data in a timely manner and evaluate the model.
Monitoring during operation and dealing with rollback	DI-5, SQ-1,6,7, PA-2,3	● Confirming the consistency of data obtained from multiple subsystems ● Consideration of customer value for AI systems ● Validation of system operation after operation ● Appropriate reflection of feedback from system users during AI system development ● Customer cooperation in AI product development ● Appropriateness of AI product configuration management ● Appropriateness of release plan ● Speed of rollback	● The amount of "seal misalignment" was confirmed to be within the normal control range during the AI system evaluation. The monitor of seal position deviation was constructed. ・The system is designed that the data acquisition status and the status of monitoring variables can be visualized during operation. ・Backup of the system, module, etc. in order to rolled back when a problem occurs.
Data privacy and security	CE-5, DI-5, SQ-2,3,4	● High security of customer data used for AI, and clarification of the scope of information disclosure and handling restrictions ● Confirm the consistency of data obtained from multiple subsystems ● Validity of risk prediction for AI products ● Ensuring the safety of AI products ● Ensure maintainability of AI products (the ability to detect, diagnose, and repair failures) ● Ensuring the safety of data fed back to learning ● Validity of the safety behavior of AI products	● After confirming the safety of the data at the customer, we reconfirmed the safety of the data. ・In addition to the function to remove anomalies in input and output data, the system shall output an anomaly message for those that cannot be removed. ・The items in the risk management table for data were updated.
Creation of a mechanism for scalability during operation	PA-8	● Relevance of the AI system update plan	● A case for a possible update of the AI system was envisaged and agreed with customers and stakeholders. An AI system update plan was developed based on those cases.
Annotations, labels, and other human teaching mechanisms	DI-2,3	● Alignment of training data with business problems ● Ensuring quality of training data sets ● Characterization of training data ● Complexity of data definitions (models to be assumed) for business issues (phenomena) ● Appropriateness of data acquisition route and management ● Whether labels and correct answer values are correctly assigned ● Independence of data used for cross-validation, generalization performance, etc.	● Define labeling rules and describe them in the detailed design document.
Data definition, model, SW configuration management	PA-3	● Adequacy of AI product configuration management ● Adequacy of release plan ● Rapidity of rollback	● Labeling by data acquisition date, time, line number, and status (e.g., abnormal) Management -Manage by "Hyper Parameter History" file. -Manage the data by revising it on the server. -It is specified at the time of making the development specification, and it is updated every time. -Associate and manage model and data. -Manage as a verification result report.

Table 7.46 Example of quality assurance study in data definition evaluation

QA Considerations	ID	Description	Examples of Case Studies
Validation of Features	DI-2	● Consistency of Data Used for Training with Business Problem ● Ensuring Quality of Training Dataset ● Evaluation of Training Data Characteristics ● Complexity of Data Definition (Assumed Model) for Business Problem (Phenomenon) ● Appropriateness of the data acquisition route and management ● Whether labels and correct answer values are correctly attached	● The features to be used were described in the learning program design document, and their appropriateness was explained to the customer. [Causal Relationships] Explain that the feature amount leads to the solution of the problem. [Property] Independence and multicollinearity were confirmed. [Cost]It is confirmed that the cost is within the computational speed to achieve the accuracy of the required specification. [Meta Level Requirement] The features to be used have been explained to the customer. It has been confirmed with the customer that there are no legal restrictions on the data to be provided by the customer. [Online] No online learning will be conducted. [Feature addition] If "film meandering" is not detected during operation, it will be analyzed and features will be added.
Data privacy and security confirmation	CE-5, DI-5, SQ-2,3,4	● Contractual arrangements regarding intellectual property rights, etc. for AI products ● Level of understanding regarding intellectual property rights, etc. for AI products ● Customer data used for AI ● Security of customer data used for AI, scope of information disclosure, and clarification of handling restrictions ● Rights of data contained in AI products ● Consistency of data obtained from multiple subsystems ● Relevance of risk prediction for AI products ● Security of AI products products ● Ensuring a control mechanism to prevent abnormal output Ensuring maintainability of AI products (ability to detect, diagnose, and repair malfunctions and anomalies) ● Ensuring safety of data fed back to training ● Relevance of safe operation of AI products	● Clearly stated in prior confidentiality agreements, memoranda, requirement specifications, etc. ● Safety of data was confirmed by the customer and then confirmed again by us. ・For abnormalities in input and output data, in addition to the function to remove them, the system shall output an abnormal message for those that cannot be removed. ・The items in the risk management table for data are updated.
Validation of evaluation data (test data)	DI-2,3, MR-1,3,6	● Consistency of data used for training with business issues ● Ensuring the quality of training data sets ● Evaluation of training data characteristics ● Data definition for business issues (phenomena) ● Complexity of data definition (model to be assumed) for ● Appropriateness of data acquisition route and management ● Whether labels and correct answers are given correctly ● Validation of learning results ● Validation of learning process ● Validation of data for verification	● Validation for 1,000 workpieces was performed by continuous operation. The amount is sufficient, but the work at the time of abnormality is artificial, so it may be insufficient, and will be added in a timely manner. The validity of the evaluation data was verified by testing.

Table 7.47　Example of quality assurance study in effect/risk assessment

QA Considerations	ID	Description	Examples of Case Studies
Selecting an Appropriate Method for AI Component Behavior Verification (Metamorphic Testing, Statistical Evaluation)	MR-4, SQ-1,6	● Validity of AI Model Structure ● Hyperparameter Recording ● Consideration of customer value for the AI system	● Validity and necessary sufficiency explained to the customer by the results of prior feature analysis, and agreement obtained. ・The threshold values of the judgment criteria were clearly indicated by displaying graphs, and the customer's consent was obtained. ・The data was divided into N parts and cross-validation was performed. ・The results are compared with the case of one variable and one feature. ● The rationale for algorithm selection and hyperparameter setting should be explained to the customer. Also, all hyperparameters shall have been tuned by training. ● Preprocessing, such as sensor noise reduction, must be performed.
Confirmation of behavior against misjudgments and unexpectedness and mapping to operational methods	DI-3, SQ-2	● Independence of data used for cross-validation, generalization performance, etc. ● Consideration of risk prediction for AI systems	● Outliers shall be excluded in P1 and input to AI components. - Outliers should be excluded in P1 and input to the AI component. ・Check the behavior of output outliers with respect to removing them. These points shall be described in the risk management table.
	SQ-2,3,4	● Consideration of risk prediction for AI systems ● Assurance of safety for AI products ● Assurance of control mechanisms to prevent abnormal output ● Maintainability for AI products (to detect, diagnose and repair failures and abnormalities) ● Maintainability of AI products (ability to detect, diagnose, and repair failures and abnormalities)	- Ensure that the AI product does not affect the system in this case. ・Evaluate the behavior at the time of abnormal value input and abnormal value output. ・In the case of abnormal output where "film meandering" cannot be determined, the system shall present what is abnormal and ask the operator to make a decision. When an abnormal value is entered in the input data, a warning is displayed.
Evaluation for prediction accuracy and execution performance	MR-1,2,5, SQ-6	● Validity of training results ● Objectives of generalization performance ● Methods to measure generalization performance ● Validity of noise tolerance (robustness) of AI ● Validity of system behavior after operation	- For learning results, prepare data and evaluation results, and explain the validity to the customer. ・Explain the generalization performance and robustness, and mention the operation guarantee after operation. ・For predictive quality, data will be visualized and performance degradation will be monitored.

Table 7.48　Example of quality assurance study in training data design

QA Considerations	ID	Description	Examples of Case Studies
Validation of training dataset	DI-*, MR-8	DI ● Feasibility of real data characteristics ● Analysis of characteristics between assumed real data model and real data	- Analysis of validity of data used for training. When cleansing is performed, the reason why it is done should be described, and the customer should be explained and consent obtained.
Annotate and check the correctness of labeled items	DI-2	● Align the data used for training with the business problem ● Ensure the quality of the training data set ● Characterize the training data ● Complexity of data definitions (models to be assumed) for business issues (phenomena) ● Adequacy of data acquisition routes and management ● Whether labels and correct values are correctly assigned	● Scrutiny of acquired and used data.
Validation of evaluation data (test data)	DI-3	● Independence of data used for cross-validation, generalization performance, etc.	● Independence of data during cross-validation and generalization performance evaluation was confirmed.
Are cleansing, padding, and data generation methods appropriate?	DI-1,2	● Ensuring the amount of data required for training the AI ● Ensuring the data used for cross-validation, generalization performance, etc. ● Consistency of the data used for training with the business problem ● Ensuring the quality of the training data set ● Evaluation of the characteristics of the training data ● Complexity of the data definition (assumed model) for the business problem (phenomenon) ● Validity of the data acquisition route and management ● Whether labels and correct values are correctly assigned	● The methods of cleansing and data generation were described in the detailed design document and explained to the customer, including the reasons for them.

Table 7.49　Example of quality assurance study in AI component implementation

QA Considerations	ID	Description	Examples of Case Studies
Validation of Hyperparameter Selection	MR-4	● Validation of AI Model Structure ● Recording of Hyperparameters	・The hyperparameter adjustment was performed using the training data, and its validity was described in the detailed design document and explained to the customer to obtain consent.
Validation of Input Data Processing (Algorithm Specific)	DI-2	● Alignment of data used for training with the business problem ● Ensuring the quality of the training data set ● Evaluation of the characteristics of the training data ● For the business problem (phenomenon) ● Complexity of data definition (model to be assumed) for the business problem (phenomenon) ● Validity of data acquisition route and management ● Whether labels and correct answer values are correctly attached	・Processing of input data (moving average, outlier removal, etc.) shall be clarified, described in the detailed design document, and the validity shall be confirmed through verification results.

Table 7.50　Example of quality assurance study in AI model development

QA Considerations	ID	Description	Examples of Case Studies
Validity of Learning Method	MR-4	● Validity of AI Model Structure ● Recording of Hyperparameters	- For the learning method, we selected a reasonable one by examining the results from time to time. The reason for the selected learning method was also explained to the customer.
Configuration management of training datasets, hyperparameters, etc.	MR-4, PA-3	● Adequacy of AI model structure ● Recording of hyperparameters ● Adequacy of configuration management of AI products ● Adequacy of release plan ● Rapidity of rollback	- Training data sets should be labeled and managed by acquisition date, time, line number, and status (e.g., abnormal). ・The model, data, hyperparameters, etc. are associated and managed. They shall be described in the requirement specification, basic design document and detailed design document.

Table 7.51　Example of quality assurance study in model evaluation

QA Considerations	ID	Description	Examples of Case Studies
Is it possible to observe and confirm changes in the internal state from learning to prediction? (DNN coverage, etc.)	MR-4	● Relevance of AI model structure ● Recording of hyperparameters	・The design should be such that the status of data acquisition and internal variables can be visualized during development. ・The data log shall be collected periodically to analyze the detection rate, and the model shall be updated if necessary.
Confirmation of actual performance against expected prediction accuracy	MR-1,2,3	● Validation of learning results ● Targets for generalization performance ● Methods for measuring generalization performance ● Validation of the learning process	● Targets and confirmation methods were set in advance, and the effectiveness was confirmed. We explicitly stated the generalization performance and improved the accuracy while taking it into account, aiming for a detection rate of 90% or higher.
Evaluation of the ability to cope with environmental dependency and explainability	SQ-5	● Explainability of AI products	● For AI products, we explained to the customer about the data used, the contents of the model, and the results, and obtained their consent. ● It is essential to develop models with the final customer explanation in mind.

7.8.3 Quality Assurance in Operations

Depending on the operational requirements, Table 7.52 to Table 7.60 show examples of operational quality assurance considerations.

Table 7.52 Example of quality assurance study in release

QA Considerations	ID	Description	Examples of Case Studies
Minimizing damage when problems occur after release	PA-3	● Release according to release plan ● Promptness of rollback	● Ver.up release time was presented to the customer. ● Backups of the system, modules, etc. are stored so that they can be rolled back when problems occur.

Table 7.53 Example of a quality assurance study in an evaluation of the effectiveness of an operational site

QA Considerations	ID	Description	Examples of Case Studies
Release interval considering adjustment and evaluation time for environment dependency	CE-1,3, SQ-1 PA-3,7,8	● Satisfaction of solving business problems with AI ● Understanding of continuous improvement for maintaining AI performance ● Understanding of continuous improvement to maintain AI performance ● Understanding of validation methods for post-operation improvements ● Consideration of customer value for AI products ● Releases as planned ● Speed of rollback ● Is the field fully satisfied ● Are updates made according to the plan	● After delivery, operation rules (monitoring method, verification method, improvement policy, etc.) should be defined and agreed with the customer. ・This time, a customer review period shall be established after delivery, and functional improvements shall be made if necessary within that period.

Table 7.54 Example of quality assurance study in monitoring

QA Considerations	ID	Description	Examples of Case Studies
Monitor inputs and outputs to check for abnormalities	CE-4 DI-5, MR-1, SQ-6,7	● Customer understanding of changes in data trends during operation ● Monitor data integrity ● Validity of system operation after operation Differences in characteristics between data during development and data during operation ● Validity of system operation after operation ● Monitoring and control of the amount of input/output data during operation	● Define operation rules (monitoring method, verification method, improvement policy, etc.) after delivery and obtain agreement with the customer. -> In this case, the following shall be done. ・When an abnormal value is entered in the input data, a warning is displayed. ・During operation, the system shall have a mechanism to manage various data with the same mechanism as during development.

Table 7.55 Example of a quality assurance study in accuracy and performance capacity monitoring

QA Considerations	ID	Description	Examples of Case Studies
Store data and logs appropriately, taking into account capacity, privacy, security, etc.	CE-4,5, DI-2, SQ-3,4	● Customer understanding of changes in data trends during operation ● Contractual agreements regarding ownership of intellectual property rights and other rights contained in AI products improved after operation and terms of use ● Contractual agreements on the ownership of intellectual property rights and other rights and conditions of use for improved AI products after operation ● Need for operational data ● Evaluation of data characteristics ● Ensuring safety of data fed back to learning ● Ensuring safety of data used for inference ● Validation of safe operation of AI products	・When improvements are made to AI products, the ownership of copyrights and other intellectual property rights included in the improvements, as well as the conditions of use, are explained to and understood by the customer, and agreed to in a maintenance contract or other agreement. ・For predictive quality, data is visualized and performance degradation is monitored. ・It was explained to the customer and agreement was obtained that the data logs would be collected periodically to analyze the detection rate, and if necessary, the model would be updated and an explanation would be provided.

Table 7.56 Example of quality assurance study in troubleshooting and maintenance

QA Considerations	ID	Description	Examples of Case Studies
It shall be reproducible in the development and verification environments	MR-2	● Define the cross-verification method during operation	・It shall be clearly defined at the time of requirement definition and risk management, and a maintenance contract shall be concluded that defines the later support system (responsibility assignment, response procedure, etc.). ・The same controller shall be used in the development environment and the actual environment, and verification shall be performed using data from the actual environment. ・All firmware ver. of the controller shall be the same. ・All the algorithm development languages should be the same. → By the above, agreement shall be obtained to guarantee the quality at the time of rollback.

Table 7.57 Example of quality assurance study in infrastructure renewal and expansion

QA Considerations	ID	Description	Examples of Case Studies
Are resources (GPU, HDD (data area), network (communication environment)) expandable as needed	SQ-7	● Monitoring and controlling the amount of data during operation	・Clearly define the policy for infrastructure renewal and expansion, and obtain consent. -> No update or expansion will be performed this time. Separate contract for system and HW renewal.

Table 7.58 Example of quality assurance study in model and training data set configuration management

QA Considerations	ID	Description	Examples of Case Studies
Configuration Management with Model, Real Configuration, and Data Combinations	SQ-7,8, PA-3,7	● Monitoring and Controlling the Amount of Data in Operation ● Hardware Maintenance According to Plan ● Software Updates ● Release According to Plan ● Release ● Quickness of rollback ● Is the site fully satisfied	● HW maintenance complies with the maintenance contract of the controller. - The model, configuration, and data are labeled and managed in combination.

Table 7.59　Examples of Quality Assurance Considerations in Incident Response

QA Considerations	ID	Description	Examples of Case Studies
Determining the necessity of investigation, modification, etc., based on probabilistic behavior	CE-8, PA-7	● Ongoing customer cooperation and involvement during operation ● Is the field fully satisfied	・Consent was obtained from the customer regarding data collection, etc., during operation.
Explanation for results/frequency that differ from inferences based on customer knowledge	CE-2,4, PA-7	● Risk tolerance for results output by probabilistic actions ● Customer understanding of changes in data trends during operation ● Is the field fully convinced	Maintain an environment that enables data collection and analysis in the event of an incident. → A data log extraction mechanism should be included for this purpose. ・ It shall be clearly stated in the maintenance contract that immediate data analysis and explanation shall be performed in the event of an incident, and consent shall be obtained.

Table 7.60　Example of quality assurance study in model update

QA Considerations	ID	Description	Examples of Case Studies
Data Addition Data Forgetting (Model Rollback) Feature Addition Training Algorithm Change	CE-3, DI-*, PA-6,7	● Understanding of Continuous Improvement for Maintaining AI Performance ● Understanding of Verification Methods for DI ● Tolerance of performance degradation in re-learning ● Reflection of experience in previous development ● Is the field sufficiently convinced	● Is it easy to change the model (rollback)? ・We explained that re-analysis and re-model development are necessary for adding data and features, and the customer understood.

References

[1] *scikit-learn algorithm cheat sheet*. https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html.

[2] SQuBOK 策定部会. “ソフトウェア品質知識体系ガイド -SQuBOK Guide-(第 2 版)”. In: 2014. ISBN: 978-4274505225.

[3] 稲垣敏之. “自動運転における人と機械の協調”. In: *IATSS Review* 40.2 (Oct. 2015), pp. 49–55. URL: https://www.iatss.or.jp/common/pdf/publication/iatss-review/40-2-06.pdf.

[4] 経済産業省. “AI・データの利用に関する契約ガイドライン 1.1 版”. In: (Dec. 2019). URL: https://www.meti.go.jp/press/2019/12/20191209001/20191209001.html.

[5] 広橋 佑紀 鶴田 浩輔 峯本 俊文. “マシンコントローラに搭載可能な AI 技術の開発”. In: **オムロングループ技術情報誌** 50.1 (May 2018), pp. 6–11. URL: https://www.omron.co.jp/technology/r_d/omrontechnics/2018/OMT_WEB_20180510.pdf.

8. Autonomous driving

8.1 Assumptions for Consideration

In the development of autonomous vehicles, AI is expected to be a core technology responsible for environment recognition, path planning, and the decision of manipulation. Expectations for AI grew rapidly in 2015 as the image recognition performance of the Deep Neural Network (DNN) exceeded human capabilities. On the other hand, DNN is concerned about black box characteristics such as difficulty in analyzing the cause of erroneous recognition, robustness such as the possibility of intentionally inducing misrecognition with minute noise that cannot be identified by humans, and unlearned data. It has been pointed out that the possibility of unpredictable behavior cannot be denied.

In Japan and overseas, with industry, government, and academia, studies on quality assurance for autonomous vehicles have already begun. However, most of the discussions have focused on the safety of the whole system. We recognize that the current situation is that concrete discussions have not yet started on how to ensure the safety of AI itself and how to guarantee the quality of AI itself.

In the QA4AI Autonomous Driving WG, we started to investigate the ideas, approaches, methodologies, and technical theories for quality assurance of AI, based on the assumption that assuring the safety and quality of AI itself is an important approach to maximize the safety of autonomous driving (AD) systems. The studies proceeded as follows.

- What kind of thinking, technologies, and methods are there for AI quality assurance?
- Formulate approaches and methodologies for quality assurance of AI-based on these
- The subject of the study is autonomous emergency braking (pedestrians) in automatic driving.

In the future, the aim is to develop an AI quality assurance theory that can be applied to more complex and advanced AD (Note 1) without limiting this study to autonomous emergency braking (AEB) based on the results of this study. At this time, the study of this QA4AI AD WG does not cover all functions of the automatic driving system and does not cover all ODD (Operational Design Domain).

(Note 1) AD defines six levels (level 0 to level 5) of autonomous driving, which are announced by the American Society of Automotive Engineers called SAE J3016. (see Table 8.1)

Table 8.1 Autonomous Driving (AD) Levels

Level	Title	Definition	Detection and response of objects/events of dynamic driving tasks
The driver performs some or all of the dynamic driving tasks.			
0	No driving automation	The driver performs all dynamic driving tasks.	Driver
1	Driving assistance	The driving automation system continuously executes the vehicle motion control subtask of either the longitudinal or the lateral direction of the dynamic driving task in a specific limited area.	Driver
2	Partial driving automation	The driving automation system continuously executes the sub-tasks of the vehicle driving control in both the longitudinal and lateral directions of the dynamic driving tasks in specific limited areas.	Driver
The autonomous driving system performs all dynamic driving tasks (when activated)			
3	Conditional driving automation	The driving automation system continuously performs all dynamic driving tasks in limited areas.	System
4	Advanced driving automation	The driving automation system continuously executes all dynamic driving tasks and responds to difficulties in limited operation in limited areas.	System
5	Complete driving automation	The driving automation system performs all dynamic driving tasks and responds to difficulties in sustaining operation continuously and indefinitely (i.e., not within limited areas).	System

Source: Partially excerpted from “Taxonomy and definitions for terms related to driving automation systems for On-Road Motor Vehicles” JASO TP18004:2018 (Society of Automotive Engineers of Japan, Inc.)

8.2 Assumed System(System to be assumed)

The functions of the automatic driving system can be roughly divided into three types: “recognition,” “decision,” and “operation.” In particular, image recognition AI based on deep learning is starting to be applied to cognitive functions that understand the surrounding environment of the vehicle. The cognitive function converts data obtained from sensors such as cameras into semantic information (vehicles,

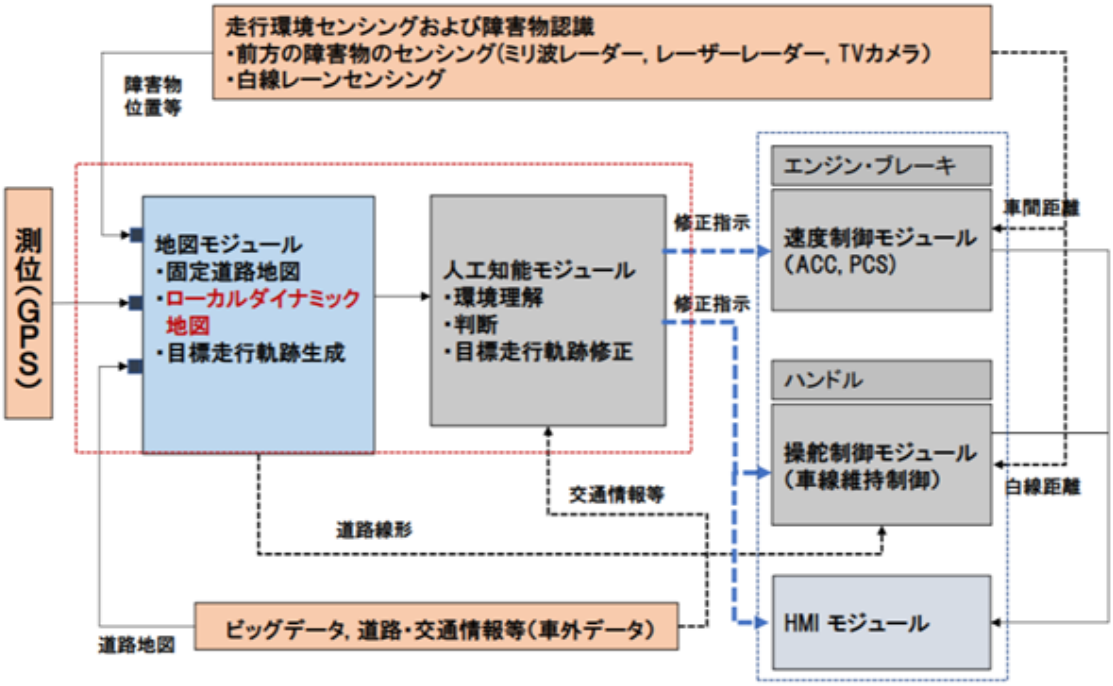
people, and the like) using image recognition AI. Then, the information is passed to a subsequent determination function as information necessary for determining the vehicle behavior. That is, the result of recognition affects decision and operation. In addition, deep learning is an AI model with a remarkable black-box property, which is an issue of quality assurance and is often used as a key technology of cognitive functions. For this reason, cognitive functions were included in the scope of this WG.

There are various applied technologies for image recognition AI used for cognitive functions, such as lane recognition (i.e., Lane Detection) and object recognition (i.e., Object Detection). The autonomous emergency braking (AEB) function is expected to be activated when an object to be avoided colliding with appears in the traveling direction of the vehicle. Therefore, in this AEB function, object recognition is a key technique for recognizing that an object in front of the eye is an object to avoid a collision.

The functions and the examination object of the AD system assumed by this WG are shown below.

- Cognitive functions of the AD system (see Figure 8.1)
- The object of the study is the object recognition function (identifies cars, pedestrians, objects, etc.) (refers to Object Detection in Figure 8.2)

Fig. 8.1　General example of autonomous driving system architecture

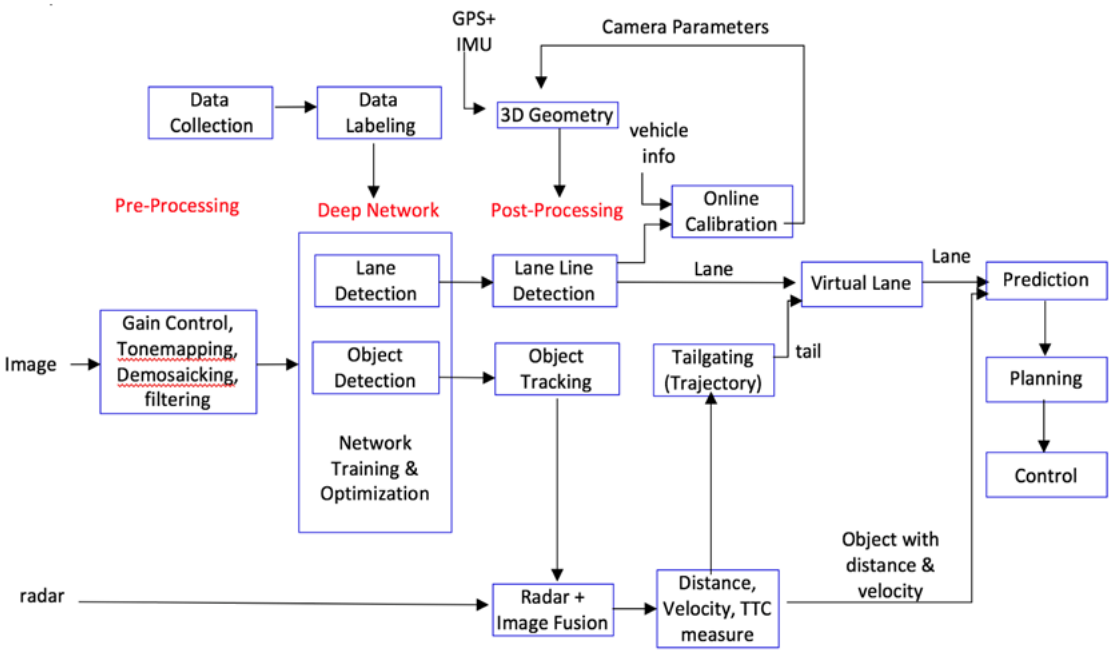


自動運転システムは5つのモジュールとセンシング部，外部通信部で構成される。縦方向と横方向それぞれ独立した出力モジュールをもち，上位の指示なしに単独で最小限の安全機能をもっており，上位からの指示に基づいて補正されることにより，信頼性や安全性の確保が図られる構成となっている。

図15　自動運転システムアーキテクチャーの一事例

引用：青木 啓二, 自動運転車の開発動向と技術課題：2020年の自動化実現を目指して, 情報管理, 公開日 2017/07/01, Online ISSN 1347-1597, Print ISSN 0021-7298, https://doi.org/10.1241/johokanri.60.229, https://www.jstage.jst.go.jp/article/johokanri/60/4/60_229/_article/-char/ja

Fig. 8.2　General example of a recognition processing flow in autonomous driving



Cited from "Perception module of Apollo project"
https://github.com/ApolloAuto/apollo/blob/master/docs/specs/perception_apollo_2.5.md

8.3 Unique issues and countermeasures

This section outlines the results of this WG study, and Section 8.4 describes the details of each characteristic of the quality assurance activity balance chart.

First, we thought it was important to analyze how the Deep Neural Network (DNN) is used in the system and worked on it as a first-step activity. Then, the "issues" and "possible measures" for quality assurance of DNN were arranged at the proposal level. This section outlines our study results.

[Features of AD system]

- Drivers and society expect AD to reduce accidents
- It may be used in various situations that cannot be assumed during development
- High mission criticality and high autonomy (Note)
- It is an embedded product

(Note) "High autonomy" means that the judgment of the model does not involve human judgment. The model refers to a machine learning model of AI.

[Assumed issues of AI quality assurance]

- Issue I: DNN may behave unpredictably for unlearned data. However, it is impossible to prepare

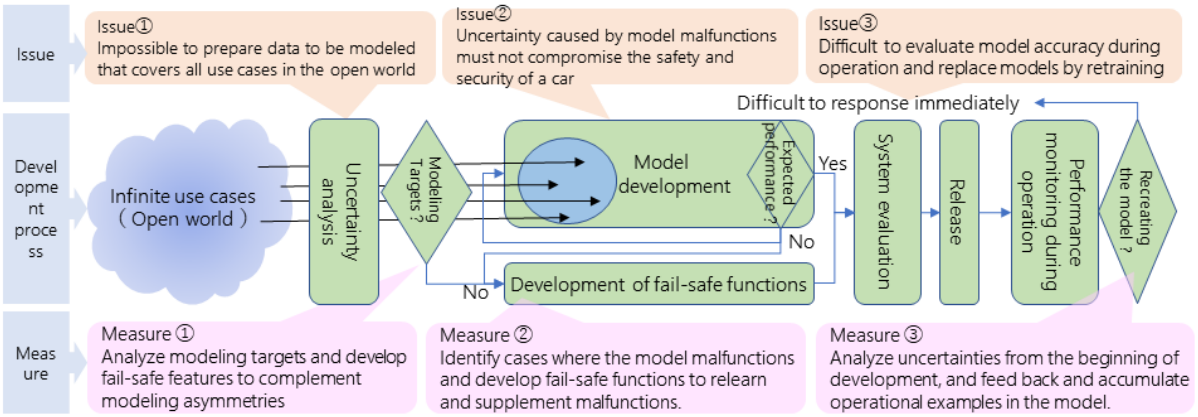
a data set that covers all use-cases during operation, and it is difficult to ensure the completeness of learning and evaluation.

- Issue II: Even if it has been learned correctly, it is impossible in principle to achieve 100% recognition with DNN, and there is a concern that erroneous recognition or unrecognition cannot be avoided. But this must not compromise the safety and security features of the car
- Issue III: There is a concern that DNN will not be able to respond to changes in the world after launch. For example, the fashion of pedestrians, the shape of vehicles, traffic conditions, and the like may significantly change before and after introduction to the market. We have to respond to these changes

[Possible measures]

- Countermeasure I: [At considering the requirements] Identify as many use-cases as possible and embody the functions required for DNN. For example, traffic scenes are analyzed by experiments and simulations to extract use-cases as much as possible. Next, the target use-cases of DNN are narrowed down, and the specifications of learning data and evaluation data are constructed. Furthermore, the necessity of developing a fail-safe function will be examined in case a use-case that cannot be handled by DNN occurs. (see section 8.4 (1) Data Integrity)
- Countermeasure II: [During model development] Theoretically and experimentally, thoroughly identify weaknesses and limitations of the DNN model, and take action on those weaknesses. For example, the use-cases that are erroneously recognized / unrecognized will be thoroughly identified through experiments and simulations, and the weaknesses will be clarified theoretically by exploring the principles of DNN. Next, we analyze the factors of erroneous recognition/unrecognition and improve the performance of the model by reviewing data construction and learning methods. If the model does not meet the expected performance, review the system requirements and develop a fail-safe function (see section 8.4 (2) Model Robustness and (3) System Quality)
- Countermeasure III: [During system operation (after-sales)] Build a system of DNN learning and quality assurance that can respond to changes in the world after-sales and maintain/improve the model performance. For example, perform an uncertainty analysis that incorporates as many unexpected events as possible during operation. A mechanism to monitor the DNN performance in the market utilizing such as OTA (Over-the-Air) and a mechanism to accumulate image data from the market will be prepared. In addition, a re-learning and quality assurance system using these data will be constructed.

Fig. 8.3　Relationship between issues and countermeasures



Next, the idea of risk reduction by uncertainty analysis is shown using Figure 8.4.

Risk extraction, clarification of modeling targets, and risk mitigation are performed in the order of (1) to (3), shown in the figure below.

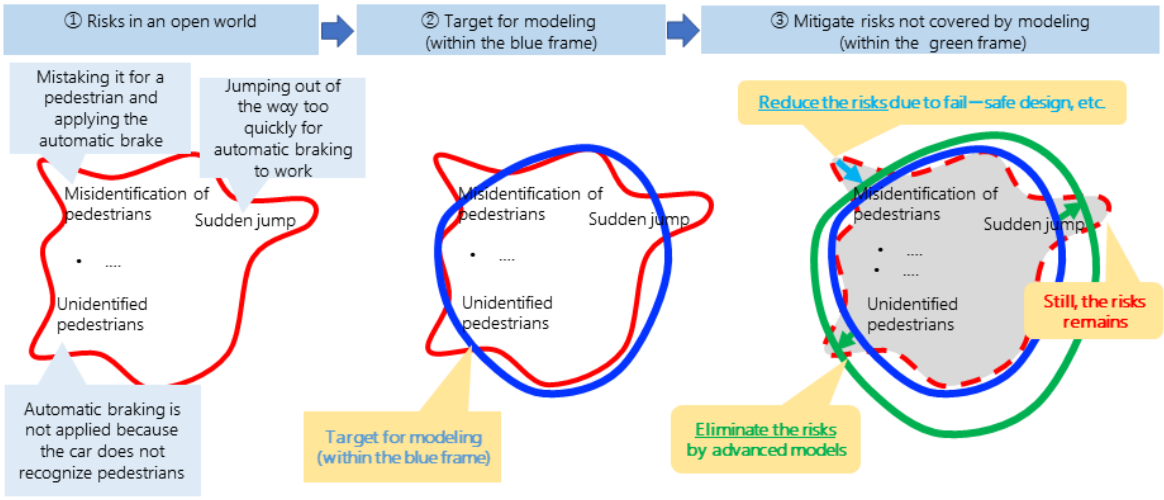
(1) Identify use-cases during operation as many as possible (Set of risks).
- Assumed risks are in the red frame of the figure.
- The projections of the red frame indicate rare use-cases that are difficult to be assumed.

(2) For the set of risks (in the red frame), the modeling target is determined (in the blue frame).
- Consider covering the set to a maximum extent, but complete coverage is difficult.

(3) Consider measures to reduce risks that are not covered by modeling.
- Eliminate risks by upgrading models
- If modeling is difficult, the risk is reduced by alternative measures (such as fail-safe design)
- Still, take various measures to cope with the possibility of remaining risks (see section 8.4 (5) Customer Expectation)

* Related information: “Classification uncertainty response level assignment” in the Appendix

Fig. 8.4　Approach to Risk Reduction by Uncertainty Analysis



8.4 Characteristics of the Balance Chart of Quality Assurance Activities

(1) Data Integrity

The first activity in data integrity is to define use-cases of the product and to determine the specification of the data. This is described below.

Since the performance of DNN is strongly influenced by the quality of the training data, we thought that it was essential to construct the data with high quality in order to assure the quality of DNN. Specifically, we thought that it was important to ensure the quality of the labels (correct values defined by Bounding-box, Segmentation, etc.) given to the images, in addition to the quality of the raw image data itself. The following is an excerpt from what data integrity aims to achieve.

- Does the data reflect the use-cases of the product (e.g., recognition of pedestrians at night)?
- Are there enough labels for the targeted recognition objects such as pedestrians and cars?
- Are there enough factors such as weather, time, area, pedestrian characteristics, and distance to the vehicles?
- Is there any bias in the factor distribution of the collected data?
- Is the accuracy of the label, such as bounding box and segmentation, sufficient?

The image data required for DNN construction for object recognition, such as pedestrians, is collected in an actual environment using an experimental vehicle equipped with sensors such as cameras. In order to do so, it is necessary to define the use-cases where the PCS is actually used, such as weather, time, location, and driving conditions, and determine the specifications of the collected data based on those use-cases. As typified by the PCS case, there are infinite use-cases in operation, such as countries where the AD system is used, traffic environment, weather, time, and the like. Therefore, it is important to extract and narrow down the target use-cases. (see Figure 8.5). In order to do so, we assumed that at least the following two steps were necessary.

Step 1: Assuming use-cases during operation as many as possible and narrowing down the requirements on DNN. In the case of pedestrian recognition, an activity plan is described below.

Define product requirements　For example, define the requirement for pedestrian recognition to be that a pedestrian 1 to 10 meters away at urban areas or intersections should be recognized when driving at 0 to 50 km/h. To define the regions where it is used as Japan, North America, China, or Europe is another example.

Design collection driving conditions　In the above-mentioned example of the above requirements, the driving plan is set so that image data can be collected in as many urban areas and intersections as possible in each region using the distance to the pedestrian and the driving speed of the vehicle as parameters. Make a plan. At that time, the driving conditions are devised so that various weather conditions, brightness conditions, and traffic conditions are included.

Plan to collect edge cases　For example, dangerous situations such as immediately before a collision are difficult to collect in normal driving tests. In addition, pedestrians wearing special clothes and special weather such as heavy rain, heavy snow, and fog are also considered edge cases. For these, it is one idea to virtually reproduce the dangerous situation on a test course or simulation and collect data.

Give label (correct value)　Give a correct value to each image for the huge amount of collected data. Using pedestrian detection as an example, generally, a correct value is given by a bounding box or a segmentation.

Step 2: Measure and refine data quality

Check the quality and quantity of raw data　Check for so-called omissions. For example, was data collected under the right conditions for a use-case for pedestrian recognition at night? Are collection factors such as weather, time, area, pedestrian characteristics, and distance to the vehicle sufficiently comprehensive? Is there a fatal bias in data distribution, such as when adults have collected enough but few children? It is considered necessary to check such things.

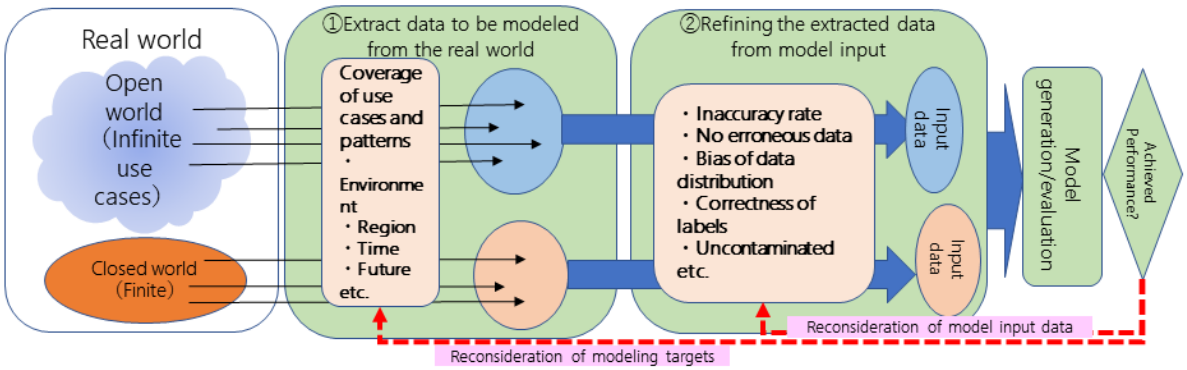
Confirm the accuracy of the label　Make sure that there are no erroneous labels, but also make sure that the way labels are applied to the same type of image does not vary. For example, for a bounding box for a pedestrian wearing an umbrella in rainy weather, it is confirmed that the way of giving the left end coordinates and the right end coordinates is consistent.

Refine data/labels　Take action on items that are found to be insufficient / problem with the above checks. Specifically, it performs operations such as data re-collection and re-labeling. What do you say is sufficient/insufficient? Standards are needed. Refining data and labels also play an important role as a countermeasure when the model performance after learning is not sufficient.

* Related information: Classification of use-cases according to “Classification of uncertainty” in the Appendix

When the inductive development of the model is repeated to improve the model performance, the above step 2 is the process of returning to the step of refining the input data of the model and re-learning. On the other hand, the process of returning to the above step 1 and reviewing the use-case targeted for the model requires plenty of person-hours, which greatly affects the development progress. However, when the use-cases during operation expand infinitely compared to the case where the use-cases during operation are finite when a response to the case that was omitted from the identification of the initial use-case was added, it is likely to return to step 1. Given this, determining the model target by the uncertainty analysis described in section 8.3 is a particularly important step.

Fig. 8.5　Two-step approach to real-world modeling data



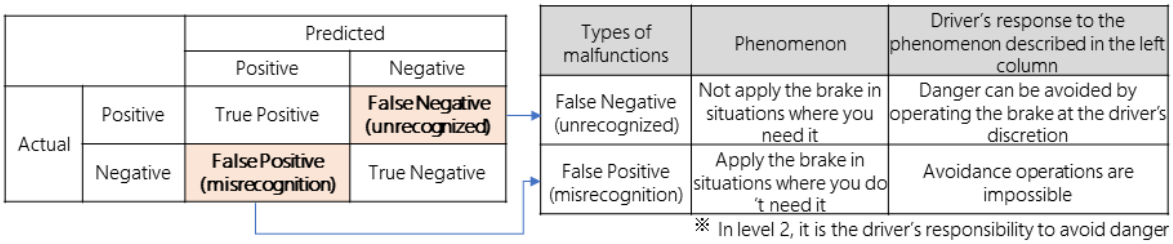
(2) Model Robustness

The indicator values for evaluating the performance of the model include accuracy, recall, precision, and the like. These are represented by a formula for computation combining each classification result of the confusion matrix. The features of the automatic driving system are shown below.

In an AD system, model recognition accuracy is an important factor related to safety. General measures for improving recognition accuracy, such as whether generalization performance is secured, whether learning has proceeded appropriately, and whether appropriate hyperparameters have been studied, must be taken as a matter of course. In addition, it is necessary to consider what kind of risk automatic driving has depended on the type of recognition error. For example, the risk that object recognition poses to autonomous emergency braking (AEB) differs between false recognition (False Positive) and unrecognized (False Negative) (see Figure 8.6).

- False recognition in AEB among false operations (False Positive) is an unnecessary operation; that is, automatic braking is applied due to erroneous recognition even when the brake is not applied. For example, the signboard on which a person is drawn is recognized as a pedestrian, and AEB is applied. If such an unnecessary operation occurs in a situation where the driver or the following vehicle cannot assume sudden braking, an unexpected rear-end collision may occur from behind.
- False recognition in automatic braking among false operations (False Negative) means that AEB does not operate in a use-case where the model has not been learned even when the brake is applied. For example, a person who has come to a costume cannot be recognized as a human. In this case, there is naturally a risk of causing a collision accident, but the risk varies depending on the range of use-cases in which the operation of the AEB is assumed and the degree of attention of the driver.
- Erroneous recognition and unrecognition generally have a trade-off relationship, and balance is important in determining performance. At this time, as in the above two items, it is considered that consideration of the specific scenarios caused by each of the target systems will be material for considering the balance.

Fig. 8.6 Differences between autonomous emergency braking (AEB) malfunctions (unrecognition and erroneous recognition)



In order to reduce these risks at the model development, it is important to take measures to improve the robustness against erroneous recognition of the model. In object recognition, not only the object to be recognized itself but also the appearance of the entire image, such as the background of the object and the sunshine conditions, affects the recognition accuracy. In consideration of mathematical diversity, semantic diversity, and social and cultural diversity existing in the real space, robustness will be improved by learning and testing models based on sufficiently diverse data.

In order to reflect the diversity of the real world in a model, not only scenes that appear on average but also scenes that are so-called edge cases (rare cases), such as those that occur rarely or are confusing, must be collected as data. No. A misleading scene is a scene where a pedestrian is mistaken for a pedestrian, or a non-pedestrian is mistaken for a pedestrian (see Figure 8.9). Note that the robustness of an AI model constructed from data, such as DNN, strongly depends on the data sufficiency in the above (1) Data Integrity, and will return to the data creation process as needed to develop (see the below (4) Process Agility).

Further, since there is a premise that an AI model is constructed based on known data, improving the robustness of the model alone is not enough to ensure the safety of the whole AD system for unexpected use-cases. It is also important to take countermeasures in various fields. For example, system-related measures such as providing a fail-safe function (see (3) System Quality) and consensus with society regarding the understanding of AI risks and explanations in the event of an accident (see (5) Customer Expectation), and the like.

(3)System Quality

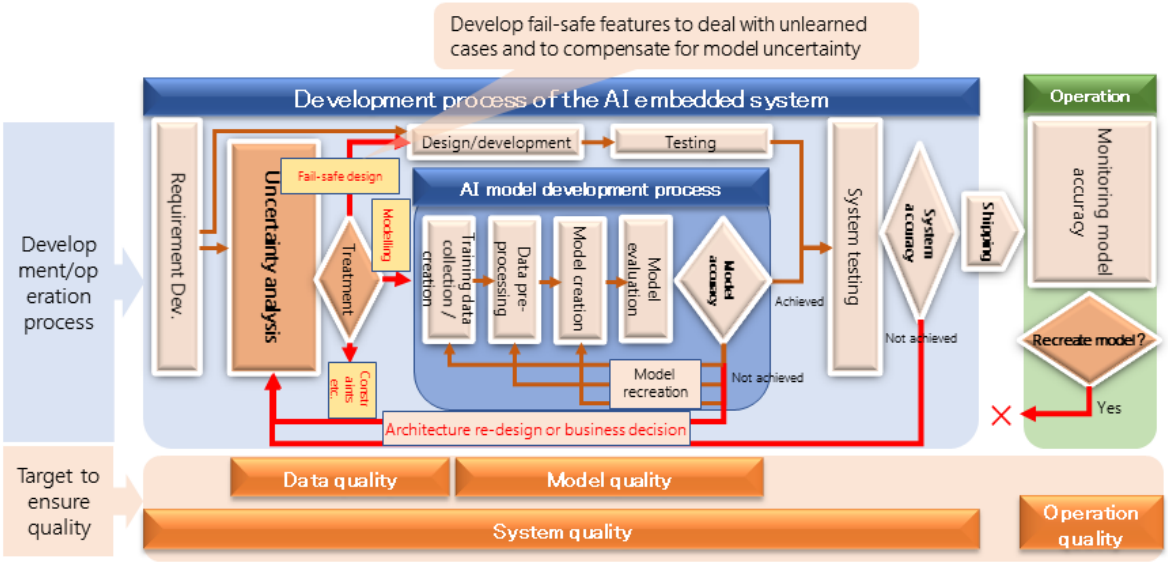
In the above (1) Data Integrity and (2) Model Robustness, it is important to improve the robustness of the model, such as improving the completeness of use-case identification during operation and preventing malfunctions. As a countermeasure to compensate for these uncertainties, ensure system quality by fail-safe design.

Fail-safe design means, for example, in AD systems, as a fail-safe function that supplements the uncertainty of use-case coverage and the uncertainty of malfunction of image recognition models, in addition to image recognition cameras, millimeter-wave radar It has a sensing function for the driving environment such as radar and a laser radar (see Figure 8.1). In addition, the safety design of the AD system is implemented by the function to judge and control the distance and speed of the target object and the information of the image recognition result by these sensors instantaneously (see Figure 8.2).

In addition, since the performance of image recognition deteriorates in heavy snow or heavy rain, it is necessary to determine whether the image recognition function is operable or in an environment and to stop the image recognition function if it is determined that operation is not possible. It is an example

of the safety design of the system.

Fig. 8.7　Development and operation processes and quality targets of AI-incorporated systems



(4) Process Agility

In Process Agility, when operating a model on a cloud service, the service update process that minimizes the impact on users is important. In this case, when a defect of the model or deterioration of the performance is detected, such an influence can be suppressed by quickly replacing the model with the improved version.

On the other hand, in the case of embedded products for which AD is applicable, it is difficult to quickly replace models after the market launch (Note 2). As a result, such products require a high degree of perfection before they can be launched to market. Therefore, in the case of embedded products, a different perspective on Process Agility will be required. That is, the agility of the iterative development process (Process Agility) due to insufficient performance of the model or system is required here.

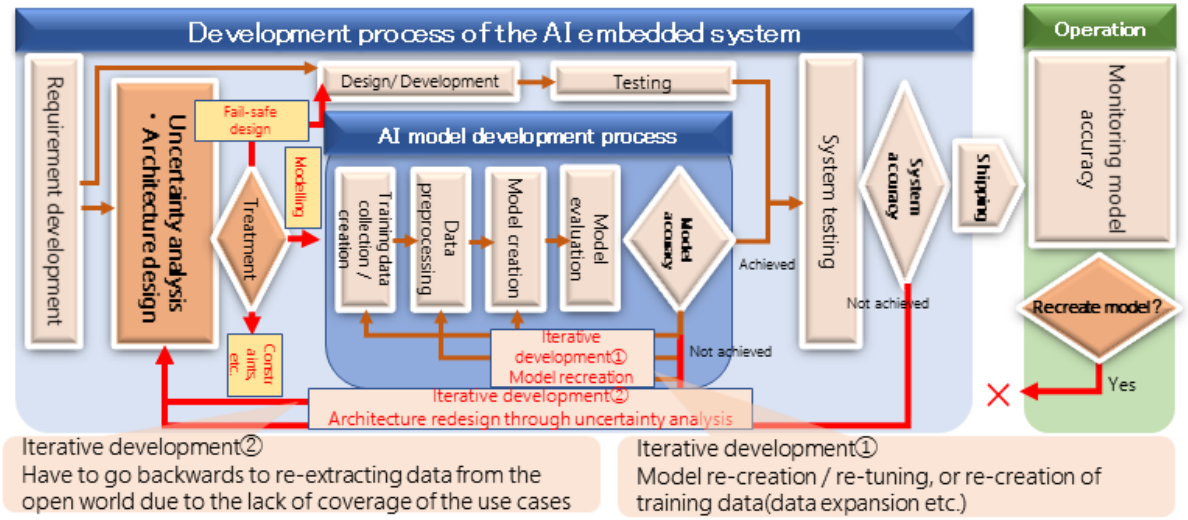
Repeated development during the development period requires agility of the following two repeated development processes. Therefore, development scheduling is performed in consideration of these.

- Repeated development 1: If the evaluation results of the developed model do not achieve the expected performance, re-tune the model or re-create the training data (data expansion, another preprocessing, and the like)
- Repeated development 2: During the development, It turns out that there is an omission in identifying use-cases during operation, and the process turns back to the use-case identification. And then, the uncertainty analysis is performed again, and it is determined whether to perform modeling or fail-safe design. In this case, the negative impact of development is significant because it affects the architecture of the entire system.

(Note 2) A mechanism for automatically replacing models while operating after a market introduction

by FOTA (Firmware On-The-Air: firmware update by wireless communication) is also being prepared.

Fig. 8.8 Two routines in the repeated development process due to underperformance of the model or system



(5) Customer Expectation

One of the goals of developing autonomous cars is to realize a safer society with fewer accidents. Creating a better society using this technology is a great mission. For that purpose, it is necessary to discuss not only technical theory but also social acceptability in terms of whether its safety is acceptable to society. In general, autonomous driving is expected to be a promising technology that will reduce accidents in total, but on the other hand, there is no denying the possibility of new risks and accidents that did not exist before. In order to prove that AD is truly safe, it is essential to accumulate long-term operational results.

The same is true for Deep Neural Network. DNN is a technology with overwhelming performance beyond human image recognition ability, and therefore can be expected to reduce accidents caused by human errors (overlooking, looking away, carelessness, and the like). However, there are issues that must be solved with the current AI technology, such as robustness and black box properties.

We believe that engineers must continue to talk with society about installing DNNs in autonomous cars while seriously working on research and development to improve the quality and safety of DNNs. We believe that it is necessary to determine the quality of AI accepted by society and the safety of AI accepted by society in agreement with society. Our major motivation is that we want to contribute to the creation of a safe and better society through the development of AI technology.

We thought it was important to address:

- Continuously seek/build on the safety of AI accepted by society
- Accumulate track record of autonomous cars equipped with AI for a long time and share it with society
- Make the limits and weaknesses of AI transparent to society
- Promote standardization of AI quality through open activities in collaboration with domestic and

international, and industry, government, and academia

In AD, the level of AD (see Table 8.1) is defined so that users can determine their own expectations for AD technology. However, since not all users can correctly grasp the contents of the AD level, the users may excessively expect AD. In order to use AD safely, it is necessary to fulfill accountability to users as follows.

- Explain the limitations and restrictions of the autonomous driving function before the user purchases the autonomous driving product.
  - It is important that users understand that AD Level 2 is not a “complete” AD but an AD “support system.”
  - Prepare insurance and explain to the user the contents of the guarantee conditions, such as driver responsibility or system responsibility, so that the user can understand AD correctly.
- When developing autonomous driving, leave the following confirmation of the development process, and make it possible to explain that the development to minimize the uncertainty factor was performed in the repeated development of models with many uncertainty factors. (See Figure 8.17 for a specific example of confirmation)
  - Complete set of model input data and model tuning results
  - Evaluation results of models and systems
  - Record of development process implementation (showing that everything has been done without omission)
- Raise awareness through activities to obtain consensus throughout society

Finally, the relationship between the problem of the automatic driving level 4 and the expectations of the user will be considered. Level 4 is defined as advanced driving automation, which is a higher and safer system requirement because the system side, not the driver, has all the responsibilities.

The discussion so far is the expectation of conservative users who place the highest priority on ensuring the safety of the automatic driving system and is a natural concept when considering quality assurance. On the other hand, the expectation of innovative (or radical) users is that they want to pursue convenience such as Autonomous Driving Level 4 even if the uncertainty risk of the AI model remains. These innovative user expectations may be the driving force behind the further development of AI. The following summarizes the differences in expectations between conservative users and innovative users.

- Conservative users expect society to demand high security for AI
  Innovative users expect society to evolve more conveniently
- Conservative users expect no negative effects of AI risk, rather than the positive effects of AI
  Innovative users expect that some negative effects may remain if the positive effects of AI are greater. That is, if this expectation is replaced with the effect of level 4 of autonomous driving, the rate of traffic accident reduction may not change as compared to level 2, but it is expected that the comfort of level 4 will be considerably greater than level 2.

Also, at level 4, it can be said that service cars such as rental cars and car-sharing are more likely to be realized than owner cars that are assumed to be owned. The reason is that in the case of an owner car, in order to guarantee the system to all users, it is inevitable to consider the guarantee from the viewpoint of the expectations of conservative users. However, in the case of a service car, operation under limited-use conditions is possible, and by combining conditions such as insurance, it is easy to realize the expected level of innovative users.

8.5 Appendix

This appendix presents deliverables in line with the work of this AD WG.

For the background of the study, we first identified the use-cases in situations where the autonomous emergency brakings (AEBs) were specified and then examined the papers on quality assurance related to autonomous driving (AD). Next, according to the classification policies of the two papers of Krzysztof' s "Classification of uncertainty" [*1] and Lydia' s "Structure of the verification goal" [*2], we classified the use-cases that were first identified and considered the relationship between the two types of classification policies. Finally, we have compiled a list of development processes, classifications of uncertainty analysis, and the items to be validated for each classification in each process.

8.5.1 Identification of use-cases

First, the conditions for the use-case of AEB were identified below.

- In autonomous driving,
- At night
- Child pedestrian
- Recognize,
- Avoid collisions
- Stop

Based on these conditions, we have washed out the use-case. (Figure 8.9)
We considered where, what, and how to protect this use-case.

- Determine assumptions for the development process
- Consider the viewpoint of quality assurance in this context

* Quality assurance, in this case, is not limited to the activities of the quality assurance department. Consider quality assurance activities in a broad sense, including building quality in the development process.

*1 Czarnecki, Krzysztof & Salay, Rick. (2018). Towards a Framework to Manage Perceptual Uncertainty for Safe Automated Driving: WAISE 2018

*2 Gauerhof, Lydia & Munk, Peter & Burton, Simon. (2018). Structuring Validation Targets of a Machine Learning Function Applied to Automated Driving: SAFECOMP 2018

Fig. 8.9　Use-case identification results

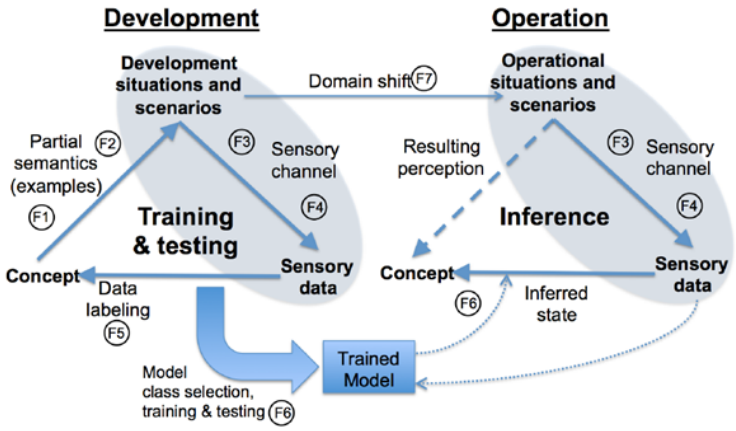
8.5.2 Survey of papers on quality assurance

Fig. 8.10 Uncertainty classification (Krzystof et al.) - Overview and process transition

A: "Categorizing Uncertainty" of perception model (K. Czarnecki et. al) ～ 7 factors (F1-F7)

Krzysztof Czarnecki and Rick Salay, Towards a Framework to Manage Perceptual Uncertainty for Safe Automated Driving, WAISE2018.

- Domain analysis on recognition using machine learning.
- Analysis of perception components using a generic model i.e., perception triangle (apexes: situation and scenarios, concept, sensory data).
- Use of two triangles, one for development and another for operation.
- 7 factors that influence perception performance are identified.
  - F1: Conceptual Uncertainty
    - Difference of interpretations by different stakeholders.
  - F2: Development Situation and Scenario Coverage
    - Degree to which the situations and scenarios used in training cover the specified concept and ODD.
  - F3: Situation or Scenario Uncertainty
    - Multiple interpretation caused by the limitation of the observability of the state of interest.
  - F4: Sensor Properties
    - Perceptual uncertainty caused by the limitation of the amount of information of interest that is sensed.
  - F5: Labeling Uncertainty
    - Uncertainty caused by mis-labeled dataset used in training and testing.
  - F6: Model Uncertainty
    - Uncertainty of the trained model.
  - F7: Operational Domain Uncertainty
    - Uncertainty of the situation, which may differ in training and in operation.



- F1,F2: Concept→Development situations and scenarios
  - F1:Conceptual Uncertainty
  - F2:Development Situation and Scenario Coverage
- F3,F4: Development situations and scenarios→Sensory data
  - F3:Scene Uncertainty
  - F4:Sensor Properties
- F5: Sensory data→Concept
  - F5:Data Labeling
- F6: Trained mode;
  - F6:Model Uncertainty
- F7: Development situations and scenarios and Sensory data→Operational situations and scenarios and Sensory data
  - F7:Operational Domain Uncertainty

Fig. 8.11　Overview of “Structuring of Verification Objectives” (Lydia et al.)

B: "Structuring Validation Targets" of perception models (L. Gauerhof et. al)　～ 3 validation targets (L1-L3)

Lydia Gauerhof, et. al: Structuring Validation Targets of a Machine Learning Function Applied to Automated Driving, SAFECOMP, 2018.

- Functional insufficiencies: deviations from the intended functionality.
  - L1: Underspecification
    - Sub Goals:
      - Environments is sufficiently well known.
      - Task is sufficiently well known.
        - Features and attributes such as color and moving situation of the subjects to be perceived are clearly defined.
      - Sensitivity against unpredictable or unspecified impact of environmental attributes is sufficiently low.
        - Decrease the Impact of situations beyond expectation to sufficiently low.
  - L2: Semantic Gap
    - Semantic gap refers to using implicit knowledge on the satisfaction of Safe Goals.
    - In the context of ML, semantic gap refers to making claims on the relevance of references used for training, test and validation data sets.
    - Sub Goals:
      - Pedestrian classes are sufficiently accurately described.
      - Location accuracy is sufficiently well described.
      - Discrepancy between real and described environment is sufficiently small.
  - L3: Deductive Gap
    - Deductive gap refers to using invalid assumptions on different abstraction levels causing an unintended functionality.
    - In the context of ML, features might be wrongly learnt or erroneously implemented.

- Measures
  - The validations targets for the deductive gap defined in the previous section can not guarantee that hazards at a system level will not occur.
  - Measures at Functional Level
    - Pre-processing of ML-input.
    - Adjustment of confidence levels in post-processing of ML-output.
  - Measures at System Level
    - Diverse redundancy.
    - Degration modes, defensive driving strategy.
    - Continuous transition between operating modes.
    - Run-time monitoring of assumptions, detection of discrepancies between distribution of environmental attributes and design assumptions.
    - System engineering review.

Fig. 8.12 Correspondence between “Classification of uncertainty (F1-F7)” and “Verification viewpoint (L1-L3)”

Relationship between Category of Uncertainty (F1 – F7) and Validation Target (L1 – L3)

Category of Uncertainty (K. Czarnecki et al.)	Validation Target (L. Gauerhof et al.)
F1: Conceptual Uncertainty	L2: Semantic Gap
F2: Development Situation and Scenario Coverage	L1: Under Specification
F3: Situation or Scenario Uncertainty	L3: Deductive Gap
F4: Sensor Properties	L3: Deductive Gap
F5: Labeling Uncertainty	L2: Semantic Gap
F6: Model Uncertainty	L3: Deductive Gap
F7: Operational Domain Uncertainty	L3: Deductive Gap

8.5.3 Quality assurance technology for recognition models

In the above (1), we identified use-cases that can be considered by our group. In the above (2), we conducted a survey of papers on the quality assurance of recognition models and obtained knowledge on “Classification of uncertainties” and “Structuring of verification objectives.” In this section, by integrating these, we defined the viewpoint of concrete quality assurance and the development process.

It will be described in the following item order.

A: “Classification of uncertainty” in recognition model
B: Classification of use-cases according to “Classification of uncertainty.”
C: “Structuring of verification goals” of the recognition model
D: Confirmation of “Structuring of verification objectives” and correspondence of “Classification of uncertainties.”
E: Classification uncertainty response level assignment
F: The positioning of uncertainty analysis and verification of verification results in the development process

A: “Classification of uncertainty” in recognition model

- F1, F2: Ideas → situation and scenario during development
  - F1: Conceptual Uncertainty
  - F2: Development Situation and Scenario Coverage
- F3, F4: Development situation and scenario → sensor data
  - F3: Scene Uncertainty
  - F4: Sensor Properties

- F5: Sensor data → Concept
  ☒ F5: Data Labeling
- F6: Model
  ☒ F6: Model Uncertainty
- F7: Situation and scenario, sensor data during development → Situation and scenario, sensor data during operation
  ☒ F7: Operational Domain Uncertainty

B: Classification of use-cases according to "Classification of uncertainty."

Fig. 8.13　Classification of use-cases according to “Classification of uncertainty.”

F1: Conceptual Uncertainty		
Pedestrian	A person moving on a road without using vehicles	・an adult ・a child ・an aged person
	A person in a wheelchair	・a person in a human powered wheelchair ・a person in an electric wheelchair
	A person riding an electric walking aid vehicle	
	A person pushing a motorcycle	
	A person pushing a two- or three-wheels bicycle (side-car is not attached and it is not pulling another vehicle)	A bicycle is a human powered vehicle consisting of two or more wheels, propelled by pedals or hand cranks
	A person passing a child vehicle	・a person pushing a baby buggy ・a person pushing a velocipede

F2: Development Situation and Scenario Coverage		
Situation of Road	・a cycleway ・a road with sidewalk ・a road with cycle path	
Situation of Environments	・display figures of shops ・windows and/or mirrors of buildings ・posters, sign boards and traffic signs ・pictures painted on cars ahead ・trees on roadside	
Situation of Target	pose	・standing ・falling ・crouching
	clothing	・wearing a hat (cap, helmet, hood) or hatless ・color and figure of clothing ・presence of reflective materials or absence ・wearing fancy dress or normally dressed
	things to carry	・carrying bag(s) ・using stick(s) ・using umbrella ・holding baggage ・holding animals ・pushing a dolly ・pushing a motorcycle, persons are on ・pushing a bicycle, person are on

F3: Scene Uncertainty	
Situation of Atmosphere	・season ・hour ・weather
Situation of Light	・lighting from front/lighting from behind ・switching state of headlights, dew condensation
Situation of Target	placed behind something or disclosed
F4: Sensor Properties	
Camera	・white balance ・angle of view ・focus ・dusts on lens
F5: Data Labeling	
-	
F6: Model Uncertainty	
Faulty Implementation	
Uncertainty in Machine Learning	
Uncertainty in Deduction	
F7: Operational Domain Uncertainty	
-	

C: “Structuring of verification objectives” of the recognition model

- L1: Reduction of risks due to insufficient specifications
  - The environment is well understood
  - Tasks are fully understood
  - Low dependence on environmental changes
- L2: Reduction of risks due to semantic disagreement
  - The pedestrian class is well described
  - Location accuracy is fully described
  - The difference between the real environment and the description environment is small enough
- L3: Reduction of risks due to inconsistency in inference
  - The training dataset is sufficient for the intended function
  - Over-training is adequately excluded
  - Under-training has been sufficiently removed
  - The key effects on machine learning are well understood
  - The machine learning function is sufficiently robust
  - Learned features are sufficient for the function
  - Parameter changes do not violate safety requirements
  - The difference between the training and operational platforms does not violate safety requirements

– Platform changes during operation meet safety requirements

D: Confirmation of "Structuring of verification objectives" and correspondence with "Classification of uncertainties."

Fig. 8.14 Confirmation of "Structuring of verification objective" and Correspondence with "Classification of uncertainty."

L1. Reduction of risk due to under-specification		Categorization
L11. Environment is sufficiently well known	Evidence-L111 hazard and risk analysis	F2
	Evidence-L112 accident database	F2
	Evidence-L113 explicit exclusion of unintended use cases	F2
	Evidence-L114 explicit assumptions on environment	F2,F3
L12. Task is sufficiently well known	Evidence-L121 requirements specification (color invariances, translation invariances)	F3
	Evidence-L122 documentation of functional modifications	
	Evidence-L123 specification of invariance and generalization attributes	F3,F4
L13. Sensitivity against unpredictable or unspecified impacts is sufficiently low	Evidence-L131 sensitivity investigation	F3,F4
	Evidence-L132 review of distributional shift	F3
	Evidence-L123 specification of invariance and generalization attributes	F3,F4
	Evidence-L134 run-time monitoring	F7
	Evidence-L135 field-based validation	F7
	Evidence-L136 statistical extrapolation	F5

L2. Reduction of risk due to semantic gap		Categorization
L21. Pedestrian classes are sufficiently well described	Evidence-L211 specification of training and validation data	F5
	Evidence-L212 evaluation of specific influences	F3,F4
	Evidence-L213 evaluation of appropriate object variations	F1
L22. Location accuracy is sufficiently well described	Evidence-L211 specification of training and validation data	F5
	Evidence-L212 evaluation of specific influences	F3,F4
	Evidence-L223 evaluation of compliance with tolerances, of size and location variation	F4
L23. Discrepancy between physical environment and description of environment is sufficiently small	Evidence-L134 run-time monitoring	F7
	Evidence-L232 degradation modes	F4
	Evidence-L233 pre-processing of ML input	F4
	Evidence-L234 evaluation of similarity between reality and specification of validation data	F5

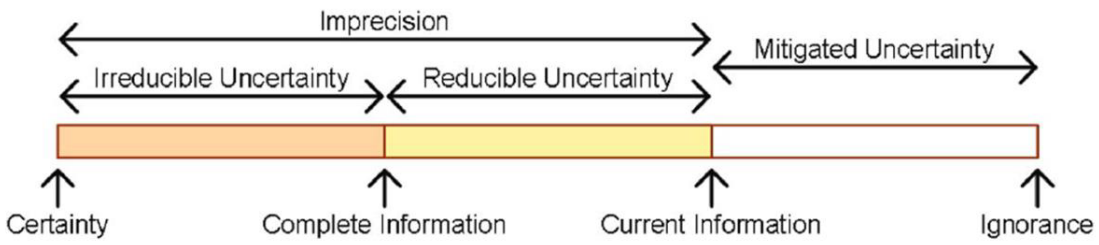
L3. Reduction of risk due to deductive gap		Categorization
L31. Data set is sufficient for intended function	Evidence-L311 evaluation of transfer of requirements to ML-specific requirements	F1,F2
	Evidence-L312 evaluation of attribute distribution within training, test and validation data sets	F5
	Evidence-L313 independence from unintended object relations	F6
L32. Overfitting is sufficiently reduced	Evidence-L321 overfitting measures	F6
L33. Underfitting is sufficiently reduces	Evidence-L331 underfitting measures	F6
L34. Essential influences on ML-function are sufficiently understood	Evidence-L341 feature visualization	F6
	Evidence-L342 correlations to features	F6
	Evidence-L343 adaptation of confidence level	F6
	Evidence-L344 uncertainty calculation	F6
L35. ML-function is sufficiently robust	Evidence-L351 evaluation of tolerance against distribution shift	F6
	Evidence-L352 adversarial attacks	F6
	Evidence-L353 statistical evaluation	F6
	Evidence-L354 evaluation of faulty inputs	F6
L36. Learnt features are sufficient for function	Evidence-L341 feature visualization	F6
	Evidence-L342 correlations to features	F6
L37. Changes to parameters do not inviolate safety requirements	Evidnnce-L371 verification specification for any changes	F6
L38. Differences between the training and target platforms do not lead to a violation of the safety requirements	Evidence-L371 verification specification for any changes	F6
L39. Changes in target platform fulfill safety requirements	Evidence-L371 verification specification for any changes	F6

E: Classification uncertainty response level assignment

Classify when the classified uncertainties should be addressed according to level (see Figure 8.15).

The levels are divided into those that are difficult to deal with, those that need to be dealt with, and those that do not need to be dealt with, and priorities are determined by taking into account the importance. Figure 8.5.3 shows the level classification for the use-cases identified this time.

Fig. 8.15　Uncertainty response level assignment



Source: Software Engineering for Self-Adaptive Systems II

Fig. 8.16　Classification uncertainty response level assignment

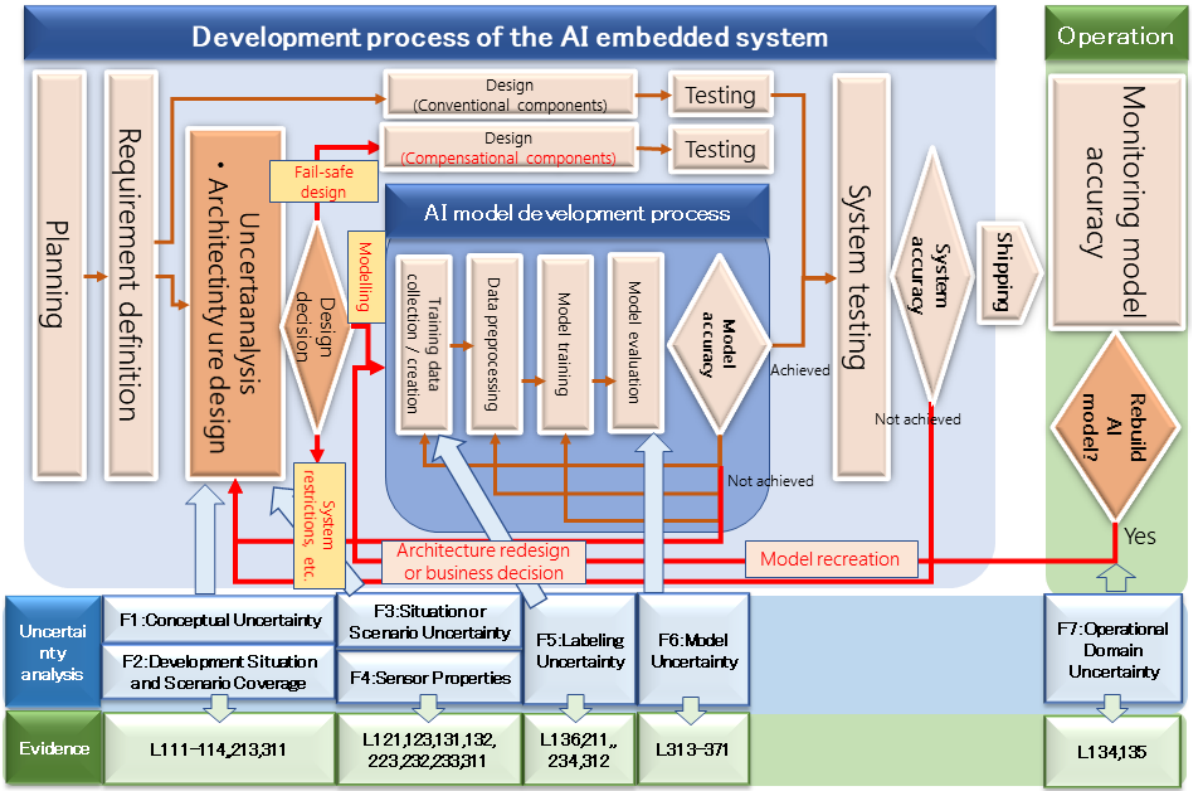
Issue: outside temperature	irreducible	reducible	mitigated
Conceptual Uncertainty	-	[few cases] Such kind of inconsistency may occur that the customer believes the car is able to run in a slight fire, which the manufacturer doesn't anticipate. Get the common consent in the stake holders.	[many cases] Range of temperature is predictable, since the temperature distribution on the earth is known.
Development Situation and Scenario Coverage	[few cases] Rare and difficult to observe phenomena relating to temperature is irreducible. Observation is characteristic in machine learnings. Nothing can be done without data.	[few cases] Unnoticed phenomena relating to temperature.	↑
Situation or Scenario Uncertainty	[few cases] Water condensation of image sensor. Method for condensation sensing and avoidance is required. This phenomena itself is irreducible.	[few cases] Data intermission when switching to backup system is reducible.	↑
Sensor Properties	-	[few cases] Malicious Physical attack, etc.	[many cases] All parameters concerning to the temperature are forecastable because of the robustness of the equipment and combination of redundancy.
Labeling Uncertainty	-	-	
Model Uncertainty	-	-	-
Operation Domain Uncertainty	[few cases] The temperature may fall out of the anticipated range in the expected operation period.	-	-

Issue: child pedestrian	irreducible	reducible	mitigated
Conceptual Uncertainty	[few cases] In defining child pedestrian, we conclude several cases are irreducible.	[middle] Disagreement of operation domain (location, hour, sunshine, etc.) between stake holders.	[many cases]
Development Situation and Scenario Coverage	[few cases] Child is not expected on the road and car behavior may change, in such situation as in snow mountain.	[middle] Particular features such as wearing prosthetic devices.	[many cases]
Situation or Scenario Uncertainty	[middle] Image disappearance caused by the light of a car in the opposite lane. A child putting on a corrugated box (system may conclude that the damage of colliding corrugated box is light).	[few cases] A child putting up an umbrella.	[many cases]
Sensor Properties	-	-	-
Labeling Uncertainty	[few cases] Impossible to remove wrong labels completely.	[none] None since found uncertainties has been removed.	[many cases]
Model Uncertainty	-	-	-
Operation Domain Uncertainty	[few cases] Change of child fashion.	-	-

Issue: wrong implementation	irreducible	reducible	mitigated
Conceptual Uncertainty	-	-	-
Development Situation and Scenario Coverage	-	-	-
Situation or Scenario Uncertainty	-	-	-
Sensor Properties	-	-	-
Labeling Uncertainty	-	-	-
Model Uncertainty	[few cases] Failures cannot be removed completely. Agreements on development process and evaluation degree provide assurance.	[none] There should not be reducible implementation errors, covered by conventional development process.	[many cases] Most model uncertainties should be mitigated, since most implementation errors are detected in evaluation phase.
Operation Domain Uncertainty	-	-	-

F: Positioning of uncertainty analysis and verification of verification results in the development process

Fig. 8.17　Uncertainty Analysis and Confirmation in AI System Development process 1



Through this uncertainty analysis, the following activities are aimed to perform. Reduction of back-tracking work due to the uncertainty of the AI model, ensuring safety that can withstand mission-critical demands, and Highly complete development of AI models due to embedded system features that make it difficult to replace models after release.

Fig. 8.18　Uncertainty Analysis and Confirmation in the AI System Development Process 2

Development Phase	Analysis of Uncertainty		Evidence of Investigation
Architecture design and uncertainty analysis of perception model	F1:Conceptual Uncertainty F2:Development Situation and Scenario Coverage	Investigate the conceptual patterns of the subjects in the system to clarify the target. Decide the design policy for the required functions whether (1) implementation using AI model, (2) use of fail-safe mechanisms to compensate uncertainty of AI model if the learning data set is insufficient, or (3) configuration of system restrictions for unimplementable functions, based on the exhaustive evaluation of the development status, use-cases and scenarios, the examination of sensors and the safety analysis.	L111 hazard and risk analysis L112 accident database L113 explicit exclusion of unintended use cases L114 explicit assumptions on environment L213 evaluation of appropriate object variation L311 evaluation of transfer of requirements to ML-specific requirements
	F3:Situation or Scenario Uncertainty F4:Sensor Properties		L121 requirements specification (color invariances, translations invariances) L123 specification of invariance and generalization attributes L131 investigation of sensitivity to environmental changes L132 review of distribution shift caused by time laps or geographical changes L212 evaluation of specific influences L223 evaluation of compliance with tolerances, of size and location variation L232 degradation modes L233 pre-processing of ML input
Development of AI model: collection and creation of learning data	F5:Labeling Uncertainty	Investigate that the labels of the learning data are correct, and the learning data set is sufficient for all use-cases.	L136 statistical extrapolation L211 specification of training and validation data L234 evaluation of similarity between reality and specification of validation data L312 evaluation of attribute distribution within training, test and validation data sets
Development of AI model: evaluation of AI model	F6:Model Uncertainty	Rebuild the AI model when its accuracy and/or performance does not achieve the development goals. Analyze the source of defects and choose the development step to go back. Further, estimate the time for completion appropriately, considering the limit of performance, deadline, etc.	L313 independence from unintended object relations L321 overfitting measures L331 underfitting measures L341 feature visualization L342 correlations to features L343 adaption of confidence level L344 uncertainty calculation L351 evaluation of tolerance against distributional shift L352 adversarial attacks L353 statistical evaluation L354 evaluation of faulty inputs L371 verification specification for any changes
Run-time monitoring	F7:Operational Domain Uncertainty	Analyze the ageing degradation of accuracy and/or performance in operation through periodical monitoring and decide to rebuild the AI model.	L134 run-time monitoring L135 field-based validation

G: Discussion of activity results

There was not enough evidence to show that the classified uncertainties supported the quality evaluation by structuring the verification objectives. However, it is highly likely that these “classification of uncertainties” and “structuring of verification objectives” are effective as a framework for organizing and relating the information of the development process, which is estimated to be enormous. Therefore, if tooling is advanced and a large amount of information can be arranged in a consistent manner, an effect in quality evaluation of learning data can be expected.

8.5.4　Automated Driving Related Standards

ISO 21448 (SOTIF: Safety of the Intended Functionality)

The safety of in-vehicle E/E systems specified in ISO 26262 is within the scope of safety against

random hardware failures and systematic failures (functional safety). However, in the case of Advanced Driving Support Systems (ADAS) and Automated Driving Systems (ADS), it is not necessarily safe just to ensure functional safety, but it is necessary to consider the lack of functions (insufficiencies of specifications and performance limitations) of the system that recognizes, judges and controls the external environment, as well as reasonably foreseeable misuse by humans. The scope of SOTIF is the safety of performance limitations, insufficient functions, misuse, and insufficient specifications of sensors, etc.

The basic idea of SOTIF is to provide safety arguments based on scenarios (i.e., interactions between the system and the external world) to determine whether there are any conditions that could cause a hazard (i.e., whether the level is acceptable). Scenarios are classified into four quadrants: unknown and known, hazardous or not hazardous. Scenarios with unknown or known hazards are identified through an iterative process, and safety is ensured by changing them to known and not hazardous scenarios through validation. The conditions that are inherent in a scenario and may cause a hazard are called triggering conditions. In the layer of the machine learning model, triggering conditions are the scenes that cause the machine learning model to mistake a non-recognized object for a recognized object (false positive) or to miss a recognized object (false negative). Safety is improved by iteratively repeating the process of subdividing the risk for each triggering condition, examining the necessity of countermeasures, and taking countermeasures.

As of September 2021, the Draft International Standard (DIS version) of SOTIF has been issued, and the International Standard (IS version) is expected to be issued in 2022.

SaFAD: Safety First for Automated Driving [6]

Released in June 2019 by a consortium of 11 companies - Aptiv, Audi, Baidu, BMW, Continental, Daimler, Fiat Chrysler Automobiles, HERE, Infineon, Intel, and Volkswagen - the Automated Driving Systems with Safety in Mind (ADS), which is a white paper that outlines guidelines for technologies and considerations for developing ADS. ANNEX B of this white paper describes the process, deliverables, and technical issues when developing a recognition system for automated driving using machine learning-based image recognition technology, and is one of the references for the automated driving part of this AI quality assurance guideline. Although this white paper states that it is "not intended to be a final statement, guideline, or standard for ADS," there is no other document that comprehensively and concretely describes the basic concept of ADS from the system level to the machine learning level, and it may be regarded as a guiding principal for ADS development.

ANSI/UL 4600Standard for Safety for the Evaluation of Autonomous Products

ANSI/UL4600 is a standard for the safety of autonomous products published in 2020 by Underwriter Laboratories (UL). Although it is not aimed for a standard for autonomous driving systems, it shows mapping of its clauses (assurance requirements) to ISO 26262 and ISO 21448, and it is expected to be used for the safety assurance of ADS. One of the main features of this standard is the extensive use of the safety case as a framework for safety assurance. The safety case is a structured description of safety arguments and their supporting evidence, which has been traditionally adopted in standards for safety critical systems. This standard provides stringent guidelines for a rigorous application of the safety case. Another important feature is the Safety Performance Indicator as a criterion for assessing safety, which is required to specify various items related to safety.

9. AI-OCR

9.1 Background and purpose of this chapter

This chapter describes the results of the working group's discussions on the concept of quality assurance in AI-OCR and the possible implementation methods at present, using the QA4AI Guidelines as a high-level document.

First, AI-OCR is defined as optical character recognition (a technology that recognizes characters from images and converts them into character codes for each character) that uses machine learning.

OCR technology has existed for a long time. It is a technology that can perform character code conversion from images by using a technique that reads images containing characters and symbols and classifies them into character codes that can be identified by a computer by comparing them with templates containing the necessary identification patterns. However, the accuracy of character recognition (the rate at which characters are correctly classified into character codes) is very low when character images that differ from the template are input, such as handwritten characters, differences in fonts, and shifts in character position coordinates. It was limited to reading specific fonts and character strings for character images corresponding to specific formats.

Against this background, with the development of machine learning technology, this technology has been applied to OCR, and the accuracy of character recognition has been dramatically improved as AI-OCR. As a result, more and more companies and organizations are beginning to introduce AI-OCR technology into their business processes. On the other hand, the concept of how to guarantee the quality of AI-OCR has not been organized yet.

Therefore, in this chapter, we will discuss the issues of quality assurance in AI-OCR and its specific problems and propose an effective approach to quality assurance for the development and introduction of AI-OCR systems.

However, it is not necessary to comply with all of them, and it is assumed to be used to decide the scope of introduction according to the target business and each company's standards. In addition, this guideline mainly describes the views on form-type OCR, which extracts arbitrary items on forms (e.g., the total amount in an invoice), and document-type OCR, which transcribes all the written text images, but it tries to include information applicable to document-type OCR as well.

As a general flow of this chapter, the 9.2 section defines the outline of the AI-OCR system, which is a prerequisite to be treated as a guideline.

In the 9.3 section, the specific issues and effective technologies of the system defined in the 9.2section are described. In this chapter, we focus on discussing issues unique to the development and implementation of AI-OCR, rather than general issues in AI development and implementation and system development and implementation.

In the **??**section, we will describe how quality assurance should be performed to solve the issues presented in the 9.3section, with actual examples.

Then, in 9.5section, we propose a quality assurance level for the development and implementation of AI-OCR systems.

9.2 Assumed system configuration

The AI-OCR system discussed in this chapter is assumed to consist of the four modules shown in table9.1.

Table 9.1 Assumed AI-OCR module configuration

#	Modules	Overview
1	Preprocessing Module	Module that performs noise removal and skew correction on the input image file
2	Character Position Detection Module	Module that identifies the coordinates of the characters to be OCRed in the image data.
3	OCR Module	Module that converts the coordinate information recognized by the character position detection module from the image to the character code.
4	Item Extraction Module	Module to extract OCR results as specific items

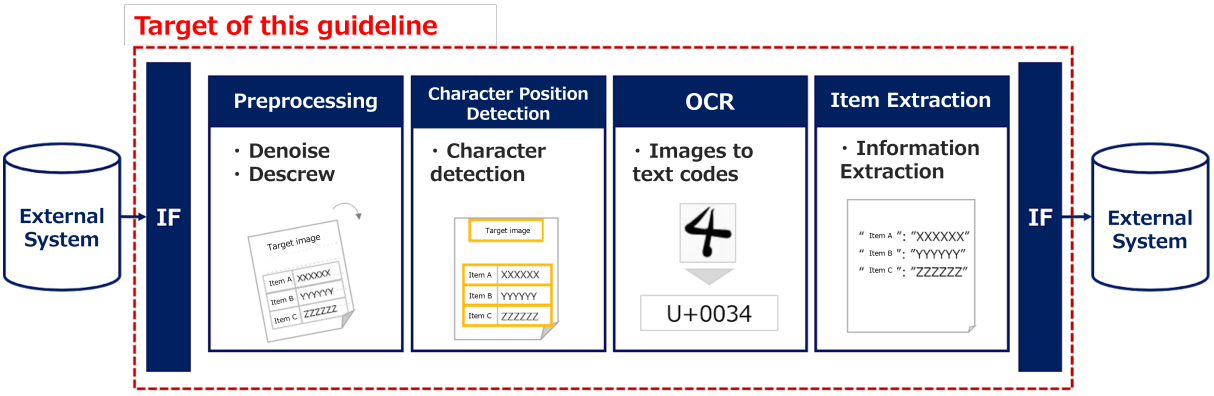
Although the interface specifications for input and output are treated as optional in this chapter, the main points of discussion for input are the extension of the input images and the file linkage method (online/batch type), while the main points of discussion for output are the output format and the linkage to peripheral systems.

These interface specifications should be adjusted according to the specifications of each system and the business to be introduced, and it is an important test perspective to guarantee the quality of the system. On the other hand, from the viewpoint of quality assurance as AI-OCR, the flow from input to output, as shown in the figure below, is the scope of this chapter.

In the next section, we will describe the details of the module presented in the figure below.

.

Fig. 9.1 Premise system and scope of the study for this chapter



9.2.1 Preprocessing Module

The preprocessing module is defined as the one that recognizes the input image file and is responsible for the image processing steps prior to performing OCR, such as removing shadows, recognizing noise, and correcting skew.

Image recognition technology is often used in this module, which recognizes shading from vectorized images, detects skew, and performs correction. An example of skew correction is to use Fourier transform to obtain the axes of the image, identify the skew, and apply a correction.

9.2.2 character position detection module

The character location detection module identifies the location of characters or text strings in the input image file. In this module, there are two approaches: the first is to use image recognition techniques to obtain the coordinates of the recognized characters in the image (machine learning approach); the second is to define the coordinates in advance based on rules (rule-based approach).

Either of these methods can serve as a character position detection module, but which one is more effective depends on the target of AI-OCR implementation. For example, if the image to be recognized is limited to a fixed layout, defining coordinates based on rules is more cost-effective and faster to implement. On the other hand, if the layout changes, there is a risk of having to redefine the coordinates. Also, if distortion or noise remains in the image due to unexpected operation of the preprocessing module functions, the defined coordinates may shift.

On the other hand, when building a character identification module using a machine learning approach, even if the forms to be recognized have diverse layouts, there is an advantage of being able to deal with them without specifying or changing the coordinate definitions by learning the layouts of various forms. On the other hand, when tackling non-generic business requirements, the process of learning models may occur before implementation, so the time to implementation tends to be longer compared to rule-based approaches.

In any of the approaches, selecting the most appropriate approach by investigating in advance whether the type, number, and layout of images to be recognized in the actual application are regularly changed is an important guideline for improving the overall system quality (System Quality).

9.2.3 OCR module

The OCR module is a module that analyzes the coordinates of the image data passed from the character position detection module pixel by pixel, classifies them into the corresponding character codes, and converts them into text data.

The OCR module is treated in this chapter as basically taking a machine learning approach.

9.2.4 Item Extraction Module

The item extraction module is a module that extracts necessary items from the information converted into text data by OCR. For example, if you want to extract only the amount of money from a receipt, this refers to the process of outputting only the total amount from the OCR result according to the output format.

As with the character position detection module, there are two approaches to this module: a rule-based approach and a machine learning-based approach.

In the case of the rule-based approach, the corresponding item can be extracted by defining the tag of the corresponding item (e.g., "total amount" = [1341,784,987,431]) at the specified coordinates.

On the other hand, in the machine learning approach, the relevant items are extracted by learning the features of the extracted items, such as words and coordinates that serve as tags for the extracted items. For this reason, it is important in the development of the item extraction module, as well as the character position detection module, to determine the approach based on the characteristics of the images that will actually be used in the business where we want to introduce AI-OCR.

9.3 AI-OCR-specific issues and points to consider

In this section, we summarize the specific issues of AI-OCR and propose measures to improve its quality. The specific issues described in this section cover the preprocessing module, the character position detection module, the OCR module, and the item extraction module. In addition, it is not necessary to apply all of the considerations listed here, but they are intended to extract which considerations should be considered according to the target business. In particular, in this section, the five quality characteristics (Data Integrity, Model Robustness, Process Agility, System Quality, and Customer Expectation) defined in the QA4AI guideline are used as the top-level documents. This document describes the specific challenges of AI-OCR and the technologies that are effective in addressing those challenges.

9.3.1 Data Integrity in AI-OCR

In AI-OCR systems, the viewpoint of data integrity is an important factor that leads to the quality of models and systems. In considering Data Integrity in AI-OCR, it is necessary to consider four characteristics of the target image: layout characteristics, character characteristics, noise characteristics, and image characteristics (see Table 9.2).

First of all, for the layout characteristics, it is necessary to consider what kind of layout the characters are described in the image file to be recognized. There are two main types of layouts to consider: regular (a layout that is always constant) and irregular (a layout that contains various shapes). Other points to be considered include the amount of information (number of characters) contained in a single image, forms containing tabular structures, the degree to which vertical and horizontal writing exists, the presence or absence of page breaks, and descriptions in a 2-up format.

These points are features that we would like to learn in advance when we take a machine learning approach. For example, it is difficult to recognize vertically written characters without learning them in a model that has learned only horizontal forms, so if there is a form image to be read that corresponds to these considerations in advance, it is necessary to properly include it in the training data.

Next, character characteristics are a consideration for the characters to be OCRed. Character characteristics can be broadly classified into printing and handwriting. Especially in the case of handwriting, it is difficult for AI to correctly recognize characters such as 1 and 7, which are difficult for humans to judge, unless it learns the specific handwriting. On the other hand, it is necessary to consider the case of printing from various perspectives. For example, for both printed and handwritten characters, it is necessary to consider character modifications such as bold and italic, font types such as Mincho and Gothic, character types such as full-size and half-size, hiragana, katakana, alphabets, kanji, symbols, and environment-dependent characters, as well as the JIS third and fourth levels of kanji. In the case of Kanji characters, it is necessary to take into account the following points of view. However, it is not realistic to build a model that covers all of these patterns, and it is necessary to define the

Table 9.2 Characteristics to consider for Data Integrity

Classification of characteristics	points to consider
Layout characteristics	Examples of layout characteristics in the target image -Regular layout/nonstandard layout -Amount of text per image -Existence of table structure -Vertical or horizontal writing structure -Page breaks -Text layout such as 1up/2up
Text Characteristics	Examples of characters to be OCRed: - Printed/handwritten characters - Modification of characters (bold/italic/underline) - Full-size/half-size characters - Character types (hiragana/katakana/English, etc.) - Symbols/environment-dependent characters - Fonts (Mincho/Gothic, etc.) - Characters that are composed of multiple kanji
Noise characteristics	Examples of noise in an image -Character blotting/letting -Light reflection -towel twisted into a headband -Background glare - Shadow reflection -Image tilt -Background noise such as copy prohibition
Image characteristics	Examples of property characteristics that define an image -Resolution (DPI) -Image size -Monochrome/color -Image brightness/saturation

necessary training data according to the items to be targeted in the actual business. In practice, if you want to OCR amounts, you only need to limit the training data to numbers, and if you want to learn complex mathematical expressions, you need to learn symbols, including environment-dependent characters. Therefore, depending on the business to be introduced and the system to be built, it is necessary to consider in advance to what extent the OCR can be used for business, or to what level of errors it can accept.

Noise characteristics are the points to be considered for noise, such as blurred characters in the image to be read. Although the causes of noise are various and difficult to specify, blurred characters, tilted images, and broken characters are also treated as noise. In addition to such unintentional noise, characters covered by a stamp or characters on a background that prohibits copying can also be considered as noise. Furthermore, in the case of forms taken by a camera, it is assumed that the background will be captured, and it is necessary to focus on background removal. Because OCR processing classifies characters based on the shading of each pixel, character cover and characters on the background may

become bottlenecks in improving the accuracy of character recognition, so it is effective to know in advance how much of such data actually exists in the population used for business. Therefore, it is good to decide the quality of the target model after discussing to what extent accurate results are required even for images that contain such noise.

For image characteristics, the considerations are image features that can be defined numerically. The main point of discussion is the resolution (dots per inch), and the dpi constraint should be considered before the project. In many cases, models are built with 300 dpi or higher on a research basis, but when it comes to actual application to business, a lower dpi may be necessary due to capacity constraints in file transfer. Therefore, from the perspective of Data Integrity, it is necessary to use the same resolution as that of the files exchanged in actual business operations for training. In addition to dpi, other considerations include image size, monochrome/color, and brightness/saturation.

As described above, from the quality characteristics of Data Integrity, four characteristics need to be considered: layout characteristics, text characteristics, noise characteristics, and image characteristics. However, it is not realistic to conduct development and quality assurance activities, including testing, that include all these considerations. Therefore, by analyzing and understanding how many of these considerations are included in the work we want to introduce AI-OCR, we can determine what kind of training data should be used to build the model and how difficult it is to build the model. Based on the results, it will be an effective quality assurance activity to define the scope of model development, define how far the model should be able to be used in business, and then implement development and test cases.

9.3.2 Model Robustness in AI-OCR

Model Robustness in AI-OCR mainly deals with the concept of measuring the accuracy derived by a model.

A robust model is defined as a model that does not become obsolete during the transition from PoC to beta development and from beta development to product development. For example, we consider hyperparameters that ensure generalization performance, reasonable accuracy metrics, and robustness to unexpected noise in real data. AI-OCR has a high output. Quantitative evaluation of the accuracy of AI-OCR is relatively easy because the output results are easy to understand. However, not only interpreting the accuracy as numerical information, but also evaluating it with multifaceted indices will help to judge whether it can be used in actual business.

The accuracy index can be expressed as a combination of the characters in the output result of AI-OCR and the characters actually output. This combination is represented by a confusion matrix in table 9.3.

Table 9.3 confusion matrix

Actual letter / Predicted letter	letter A (Positive)	letter B (Negative)
Character A(Positive)	True Positive: TP	False Negative: FN
Character B (Negative)	false Positive: FP	true Negative: TN

Typical accuracy indices used are the percent correct, percent fit, percent repeatable, and F-measure. The characteristics of these metrics are shown in Table 9.4.

The above metrics are only for the entire text and only represent an evaluation of the model itself.

However, when introducing AI-OCR into the business, it is important to evaluate the accuracy not only in terms of characters, but also in terms of items (number of items correctly answered with an

Table 9.4 Typical Accuracy Metrics

perspectives	Metrics	Formulas	Features
character evaluation	Percent Correct	(TP+TN)/ (TP+TN + FP + FN)	Valid (easy to calculate) for simple OCR module accuracy evaluation
	Acceptance rate	TP/(TP+FP)	There are few misclassifications, but there are many omissions (percentage of values that cannot be obtained)
	Reproduction rate	TP/(TP+FN)	Many misclassifications but few omissions
	F-value	(2*fit rate*repeat rate)/ (fit rate+repeat rate)	used to evaluate overall accuracy from preprocessing module to item extraction module
Efficiency Assessment	Levenstein distance	For each character read from the correct string Calculate the editing distance based on the number of insertions, substitutions, and deletions for each character	used to estimate how much time it will take to correct the OCR result if it is wrong
	per-item accuracy	Number of items correctly answered with an exact match / Number of items to be read	Percentage of items that AI-OCR can read without error in the business

exact match/number of items to be read). By using this indicator, we can use it as a guide to estimate the effect of business reduction and ROI. However, the accuracy of each item varies greatly depending on the item to be read. For example, if you want to read an address as an item, the number of characters to be read is about 10 to 20, but if you want to read a person's name as an item, the number of characters to be read is about 2 to 7. Therefore, the accuracy per item is likely to be lower for addresses with a larger absolute number of characters, and it is important to evaluate the accuracy based on the characteristics of each item.

In addition, the Levenshtein distance may be adopted from the perspective of the work reduction effect. The Levenshtein distance is a method to measure how far the output result from AI-OCR is from the actual correct answer data in terms of insertion, substitution, and deletion of characters, and it can be a very effective evaluation index in the case of a workflow where the results are read by AI-OCR, corrected by humans, and then input into the system.

As we have mentioned, there are various indicators of accuracy, and it is important to understand their characteristics and measure them from multiple perspectives for the issues we want to solve.

On the other hand, the data (test data) from which accuracy is derived is also important. In considering the test data, the ratio of training data, test data, and cross-validation data become important. In addition, it is necessary to consider the various forms used in the business for the test data, and to make sure that the data is not evaluated using biased data or data with almost the same layout. In addition, it is also necessary to check the ratio of noise in the test data and the mixture of forms with high frequency of occurrence in business and forms with high degree of difficulty. By using multifaceted indices to measure test data that has been thoroughly examined, it is possible to estimate the quality of the model

and its feasibility for business applications.

In consideration of Model Robustness, whether or not the accuracy of the model is stable when it is continuously trained can also be an evaluation perspective. Therefore, by checking the degree of improvement in accuracy when training data is added, the availability during operation becomes clearer. In addition, in Model Robustness, it is also important to analyze the causes of AI-OCR errors and understand the error trends as a quality assurance activity. If we understand whether reading errors are caused by the character position detection module, the OCR module, or the item extraction module, the characteristics of the model and the risks after its introduction will become clearer.

9.3.3 Process Agility in AI-OCR

We will examine Process Agility in AI-OCR systems from two aspects: the first is agility in model development and the second is agility in the operational phase.

The first is model development agility, and the second is operational agility. The main issue of agility during model development is mainly focused on the data part.

AI-OCR is incorporated into various business processes, but due to the nature of paper forms as AI-OCR targets, secure data containing personal information may be included. Therefore, when outsourcing the construction of AI-OCR models, it may be necessary to mask personal information. In addition, when creating data for training and testing from image data, transcription of the text is essential. Therefore, if the rules for handling the collected data are not decided in advance, there is a risk that the start of development will be delayed and mobility will be reduced.

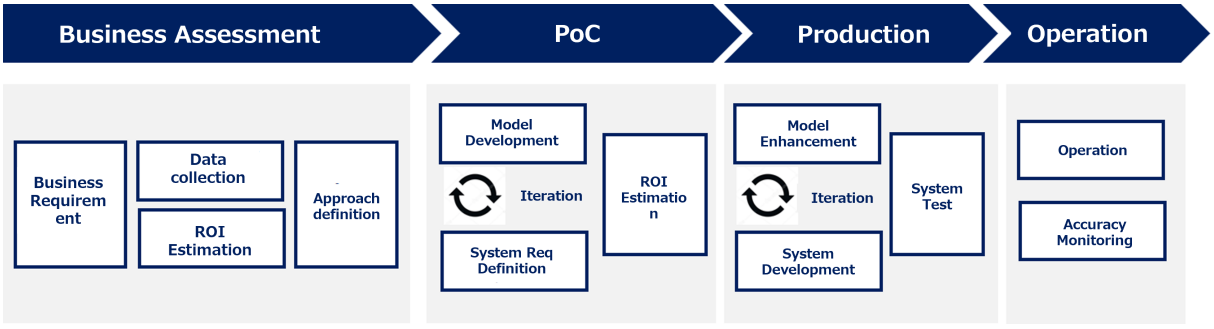
If there is a lack of consideration during the development of ROI calculations, target accuracy, and the data on which accuracy is based (e.g., whether or not to use handwritten data), the conditions for completion of model development cannot be defined, and model development may not be completed.

As for mobility during operation, it is necessary to establish a mechanism to constantly monitor the accuracy of the system. The key to improving mobility is to build a system that can quickly detect a decline in the accuracy of the model and modify the model with minimal impact on the business.

In this section, we advocate the development process shown in the following figure when integrating AI-OCR into business.

.

Fig. 9.2 The development process for AI-OCR systems



First of all, in the introduction study and assessment phase, we will select the operations that are considered to be effective for AI-OCR, and proceed to organize the operations after the introduction of AI-OCR, as well as determine whether or not to procure data and make adjustments. At this stage, it is recommended to directly check the data to be used in actual operations. Then, based on the data to be

used in actual operations, define the approach to determine whether the coordinate specification type or the form recognition type is more effective.

In the model development/requirements definition phase, it is recommended that system requirements definition and model development be done in parallel. The purpose of this is to organize the requirements to be realized and the actual model to be built, and to organize the overall picture when building the system. Therefore, it is necessary to constantly check the status during the model development and fine-tune the requirements based on the results of trial and error to improve the feasibility of systemization. In particular, because AI-OCR applies image recognition technology, it is necessary to examine whether the model to be built at the time of development can be operated in an environment that can be prepared by the company. Once these tasks are completed, we will make a decision on whether to actually develop the system. At this point, the decision is made not only based on the accuracy of the model, but also on whether it can be incorporated into the actual system, in light of the requirements definition.

In this development phase, the system development is carried out in parallel while improving the performance of the model by setting model parameters, adding rule-based corrections, and adding training data to address the issues that arose in the model development/requirement definition phase. When these are completed, the system will be tested/released and moved to the operation phase.

In the operation phase, we will continuously monitor not only the operation of the system, but also the accuracy of the system, so that we can identify forms that cannot be read by AI-OCR during operation and continuously improve the accuracy. Also, when the model is updated, it is important not only to measure the recognition rate but also to check the accuracy of forms that could be recognized before the update.

9.3.4 System Quality in AI-OCR

In developing and implementing AI-OCR, general system development methods are of course necessary, but in this section, we will focus on issues specific to AI-OCR.

AI-OCR is designed to recognize various languages and characters, but in some cases, it needs to be corrected by a dictionary. Especially for business terms and standardized words such as company names and addresses, dictionary correction is effective in improving the overall quality of the system. These dictionaries are rarely constant, and in the case of addresses, for example, changes in addresses occur due to the consolidation of municipalities. Therefore, the system needs to include a mechanism to update these dictionaries on a regular basis.

As a characteristic of AI-OCR related to these dictionaries, it is also necessary to respond to social changes such as changes in laws and regulations. For example, it is important to register dictionaries for the new year's issue and to consider format changes due to changes in tax rates.

In addition, when incorporating AI-OCR, it is difficult to process all forms with 100

In terms of incorporating AI-OCR into business processes, it is important to determine the infrastructure configuration based on the business time and the number of forms that need to be processed. It is also necessary to consider whether the target business requires online processing or nightly batch processing when designing the system infrastructure.

9.3.5 Customer Expectation in AI-OCR

The purpose of developing and introducing AI-OCR is defined as the contribution to work substitution, work speed and work quality, as shown in Table9.5.

First, let's talk about work replacement. When developing and implementing AI-OCR, the level of

Table 9.5 classification of expectation in AI-OCR

Expectation item	expectation depth
level of work alternative	complete replacement of user's work
	User task support
	used as reference information (for storage purposes)
work speed level	faster than user
	equal to users
~~Level of work quality~~	AI makes mistakes that humans can't make (homogenization is possible)
	Quality of work differs from person to person

work replacement required will contribute to the final quality. The level of work replacement can be divided into complete replacement (AI-OCR replaces all the work), work support (AI-OCR supports normal work), and use as reference information (AI-OCR converts the data into electronic data and stores it). The quality of the final model required will vary depending on the degree of this purpose. When starting development, it is recommended to consider the quality to be achieved according to this work alternative level.

As for the work speed, it is also necessary to consider whether the development and introduction of AI-OCR requires a work speed (i.e., processing speed) comparable to that of a human, or whether it needs to process faster than a human. It is also important to consider whether the infrastructure to realize the processing speed to meet the expectation can be prepared.

In developing and implementing AI-OCR, it is also important to improve the level of work quality. This is because, while human work varies from person to person, AI-OCR can make mistakes that cannot be made by humans (e.g., misrecognition of "=" and "d"). It is also necessary to consider how much the detection and correction of these errors will affect the business. In addition, there are some technologies that allow people to detect these errors in advance, such as the use of confidence, and it is necessary to consider the introduction of such technologies.

9.4 Example of AI-OCR application of quality assurance technology

9.4.1 Example of quality assessment applying metamorphic testing

As a method to evaluate the quality of the developed AI-OCR from the viewpoint of Model Robustness, we propose a test design that applies Metamorphic Testing (hereinafter referred to as MT) as described in 3.3.2. First, we describe two reasons why MT is effective for evaluating the quality of AI-OCR. 1) AI-OCR requires high Robustness. The forms inputted to AI-OCR have various characteristics due to the nature of the customer's business and the person who fills in the forms. AI-OCR with high robustness can recognize forms with any characteristics with high accuracy, which can be called high quality AI-OCR. (2) Efficient identification of weaknesses and defects in AI models Since AI models are created inductively from training data, it is very difficult to verify them by defining input-output relationships based on an understanding of their internal logic. Therefore, MT is applied to perform validation by repeatedly comparing the output results among various test cases. Identifying what kind of changes in input data are vulnerable leads to efficient identification of weaknesses and defects in the model. When defects or model weaknesses are identified, countermeasures such as tuning of training parameters, identification of data to be retrained, and review of internal logic are considered. For these

two reasons, MT to check the robustness of the model is effective in evaluating the quality of AI-OCR. We can evaluate the quality of AI-OCR in terms of Model Robustness by creating and testing various test cases using metamorphic relations. However, since countless patterns of metamorphic relations are possible, the number of test cases can be unlimited. In order to obtain the maximum verification effect with a realistic number of test cases, it is necessary to devise a way to add changes to forms using metamorphic relations (Reference [1]). Therefore, it is desirable to narrow down the test cases and consider the priority by using the obtained viewpoints for transformation based on the metamorphic relation according to the following derivation example.

<Example of Perspective Derivation (1) Identify recognition patterns that conventional OCR without AI is not good at. (2) Analyze the characteristics and frequency of forms generated in the customer's business. (3) Identify the patterns where misrecognition has a significant impact on customer operations.

For example, let's say that we investigate (1) above and find out that the problem of conventional OCR is "the recognition accuracy of the case where the ruled line of the item column and the character overlap is low". In order to confirm whether AI-OCR can overcome this problem, a form (1) in which ruled lines and characters do not overlap and a form (2) in which ruled lines and characters are intentionally overlapped are created by adding changes to the form (1) using the transformation of the metamorphic relation called Noise-based (transformation of input that does not affect the output result) (Reference [2][3]). (Reference [2][3]), and create a form (2) that intentionally overlaps ruled lines and characters. Then, by comparing the recognition results of Form 1 and Form 2, we evaluate whether AI-OCR is able to overcome the conventional problems. In addition, we investigate (3), and assume that there is a pattern in which misrecognition greatly affects the customer's business: the recognition of monetary items. In this pattern, it is necessary to recognize with high accuracy handwritten numerals that are close in shape, such as "1" and "7". First, we prepare a form with a handwritten "1" in the amount field, and then, using the metamorphic transformation called "Heuristic" (which changes the input to be closer to the original data), we create multiple forms with the font changed to "7" which is closer to "1" (Reference [2][3]). Then, we compare the recognition results among the created forms, and evaluate whether the handwritten characters "1" and "7" can be clearly distinguished with emphasis and priority. It is also effective to make use of such a high-priority test perspective in creating test cases using metamorphic relationships. As mentioned above, the consideration of effective metamorphic relationships is an important point in test design applying MT.

[1] J. Nakagawa, "Efforts to advance quality evaluation/testing of software with Deep Learning," JaSST'19 Hokkaido, 2019. [2] C. Murphy, et al., Properties of Machine Learning Applications for Use in Metamorphic Testing, SEKE2008, 2008. [3] C. Murphy, Applications of Metamorphic Testing, 2011, http://www.cis.upenn.edu/ cdmurphy/pubs/MetamorphicTesting-Columbia-17Nov2011. ppt

9.4.2 Case Study on Quality Assessment Using Form Item Analysis

In AI-OCR with item extraction, it is necessary to perform a quality evaluation considering the characteristics of each item. Therefore, when evaluating quality from the viewpoint of Data Integrity, it is necessary to consider not only the aforementioned four characteristics (layout characteristics, text characteristics, noise characteristics, and image characteristics) that are unique to AI-OCR, but also the characteristics of each extracted item of a form.

The AI-OCR WG defined the items that are often the target of item extraction in AI-OCR as standard item characteristics, and the items that should be considered for each form as form-specific item characteristics (see Table 9.6).

The standard item characteristics define items that can be included in basically any form and can be

Table 9.6　Items targeted for extraction in AI-OCR

Property Type	Overview	Examples
Standard Item Characteristics	Any Form May Contain, Characteristics to Consider in Common Items	Amounts and dates are used in a variety of forms such as invoices and account transfer requests.
Form-specific characteristics	Form-specific items that can be listed separately for each form, or characteristics that should be considered for the entire form.	slip numbers, etc.

the target of item extraction. The standard item characteristics define the characteristics that should be considered for "amount," "date," "company name," and "phone number. For the form-specific characteristics, the points to be considered for each form were defined for "invoice," "questionnaire," "my number form," "bank transfer request form," and "detailed line of form. We plan to add examples of these form-specific characteristics through AI-OCRWG in the future.

By referring to these characteristics, we believe that we can help quality assurance activities by satisfying the perspectives for creating test data for each item of forms.

9.4.3 Standard Item Characteristics

The items defined as standard item characteristics, such as "amount," "date," "company name," and "phone number," are often the target of item extraction in many forms. Therefore, we set these items as standard items, and the following table shows the perspectives and examples that should be considered in creating a test oracle for AI-OCR.

For example, even a simple item like "Amount" has as many as ten characteristics to be considered, and we believe that we can create appropriate test cases by understanding which characteristics the data to be read in actual operation falls under and preparing appropriate test data.

total amount

- With/without ¥ mark (e.g. ¥ 10,400)
- Space between the ¥ symbol and the amount
- Notation behind the amount (e.g. yen, -, 也, etc.)
- Comma-separated (e.g., ^{3} ,000,000)
- box separator

Quality Level	Data Sets	Metrics	Status
Level1 : Experiment Level	Data is collected without consideration of data unbalance (just sampling)	Single metrics	The performance of the model cannot be guaranteed, but it is ready to be tasted and evaluated.
Level2 : Optimized model Level	Data is just sampled one. But, the model is evaluated by train/test/val datasets	Several metrics	The performance of the developed model itself has been evaluated, but whether it can be used in actual applications cannot be determined.
Level3 : Business fitting Level	Untrained data applied in actual operations, after satisfying the perspectives of Level 2.	Several validated metrics	A state in which it is possible to evaluate that the developed model performs as well in business.
Level4 : Business efficient Level	After satisfying the level 3 perspective, the accuracy is evaluated using data equivalent to actual operations.	Evaluate business efficient ratio	A state in which it can be evaluated that operational efficiency can be improved during actual operation.

- minus sign notation (e.g., -100,000)
- Notation in foreign currencies such as dollars and euros (e.g., € 800,00)
- Indication of tax excluded or included (e.g., ¥132, 220 (excluding tax))

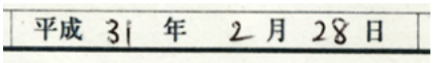
- Preprinting of units such as "hundred" and "thousand

- Kanji description (e.g. 也 650 yen)

date

- with year zero-fill (e.g., 2020/04/02)
- without year zero-fill (e.g., 2020/4/2)
- with year/month/day filled in (e.g., April 2, 2020)
- Japanese calendar with zero-fill (example: April 22, 2020)
- Japanese calendar without zero-fill (e.g., April 22, 2020)
- Source code omitted (e.g., April 22, 2020)
- Japanese calendar with original year abbreviated (e.g., R02/04/22)
- Japanese calendar with original year notation (e.g., May 1, 2019)
- Selectable year (checkmark or circle)

- Preprint "Gen-issue", "Year", "Month" and "Day".



company name

- Branch name/store name (e.g. Times Daiei Tondabayashi)

phone number

- TEL description (e.g. TEL: 08012345678)
- Sequential number with at least 10 digits (e.g.: 08012345678)
- With hyphen (e.g. 080-1234-5678)
- With () (e.g. 080(1234)5678/(03)12345678)

- With fax (e.g. 024-1234-5678 (FAX))
- Representative number (e.g., (03)(substitute)1234-5678)
- 8-digit number, including area code abbreviation (e.g., 1234-5678)

9.4.4 Form-specific characteristics in invoices

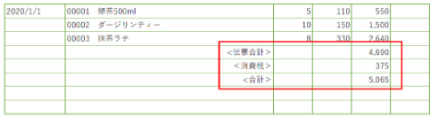
Invoices are often the target of AI-OCR, and many products dedicated to reading invoices are available.

On the other hand, it is necessary to create appropriate test cases and evaluate whether the model can be used in actual business.

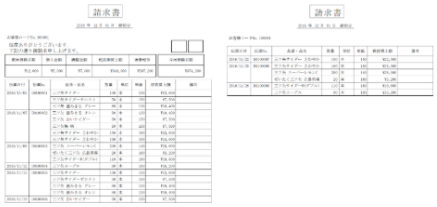
In this section, we have listed the unique characteristics of invoices, excluding standard items, that should be considered. By identifying the extent to which the invoices to be read in actual business operations cover the characteristics to be considered, we can use them to create or thin out the test cases. The characteristics specific to invoices are listed below.

invoice as a whole

- Distortion/noise due to the use of faxes
- Inclusion of information in the statement (e.g., total in the statement line)



- multiple pages (with different formatting from page 2 onwards)



form name

- with or without character spacing

- Enclosing text

itemize

- with or without shading

品目	単価	数量	金額
商品 1	15,000	5	75,000
商品 2	20,000	4	80,000
商品 3	9,000	6	54,000
商品 4	3,000	7	21,000

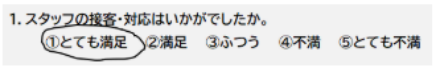
9.4.5 Form-specific characteristics in questionnaires

Questionnaires are often handwritten, and the process of tabulating the results is time-consuming. Therefore, they are often subject to AI-OCR. The considerations for AI-OCR in questionnaire documents are described below.

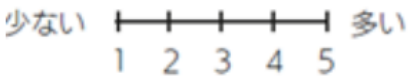
In addition, the characteristics of questionnaires are such that there is a wide range of formats that can be defined by the company, and it is assumed that this information can be used to define a format that is easy to read by AI-OCR after understanding the considerations listed in this section.

rounded selection

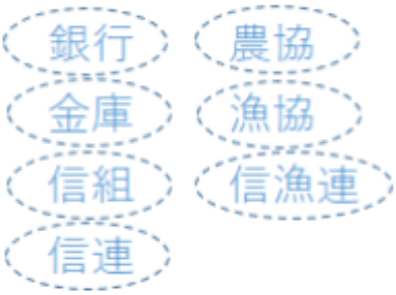
- Enclose the entire target or just a number



- A checkmark for the circled item

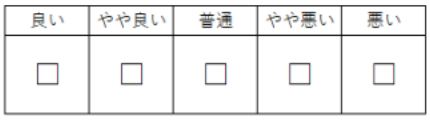


- Surrounded by dotted lines



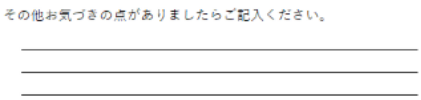
select checkbox

- Checking method (tick, fill, 0, ×, line only, etc.)



free description

- Consideration for multi-line descriptions



- Consideration for overflow description

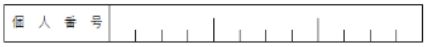
9.4.6 Form-specific characteristics in My Number forms

In accordance with the My Number Law that came into effect on October 5, 2015, the entry of the My Number has been added to forms and various application documents handled by local governments. In this section, we define forms in which the entry of the My Number is mandatory as the My Number forms.

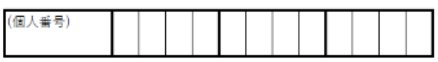
The format of the "My Number" forms tends to depend on each local government, and the following is a list of considerations for the variations in format.

my number

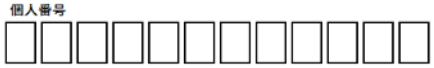
- Line separator for numbers



- Short bold separator for every four characters



- Separate boxes by one character



- gray background

name

- (Mr.) and (Ms.) in the entry field

9.4.7 Specific characteristics of the form in the transfer request form

A bank transfer request form is a form used to process bank transfers at banks and other financial institutions. In the context of improving the efficiency of paperwork in the financial industry, bank transfer requests are often the target of AI-OCR reading.

On the other hand, the layout of money transfer request forms differs among financial institutions, and there is a large variation even among similar items. Therefore, the following are some points to consider for money transfer request forms.

entire

- Consideration of the handling of the money transfer receipt as it is included in the set

name of payee financial institution

- Selectable financial institution

- in the same frame (series)

- in the same frame (parallel)

- separated by commas (e.g., Yamaguchi Bank, Waki Branch)

deposit type

- Rounded selection (e.g., circle one of the following: Regular, Checking)
- Checkmark type selection (e.g., ☐ ordinary ☐ current)

account number

- Symbol number (e.g. symbol/12345 number/12345678)
- Five- or six-digit old number (e.g. 12345)

address

- Set entry of phone number

9.4.8 Form-specific properties of detail lines in forms

Many forms contain detail lines, and reading the detail lines is one of the most difficult tasks for AI-OCR. The reason for this is that details often contain multiple items in a complex manner, and the variance of the format is often very large. Therefore, the characteristics of the statement lines in the forms described here are not based on the target items described so far, but on the approach to list the characteristics that should be considered when reading the statement lines from the risk that may occur when reading them with AI-OCR.

characteristic of detail lines that degrade character recognition accuracy

- misrecognition of characters due to border overlap (overlap with pre-printed lines)

- Misread the border (misread the dotted line separator as 1)

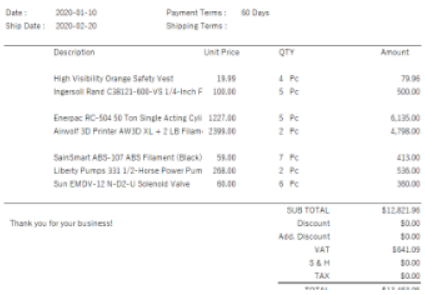
数量	金額
15	3,000

characteristic of the detail line that causes failure to associate target item with a value

- group suppress

商品区分	商品コード	得意先コード	日付
1	001	0001	05/01
			05/02
		0002	05/01
	002	0002	05/03
			05/04
2	002	0001	05/01
	001	0001	05/01

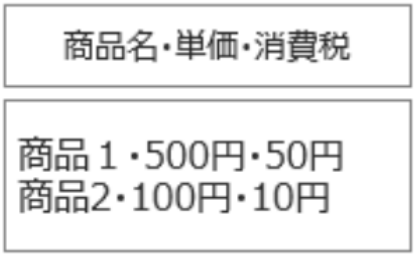
- without lines



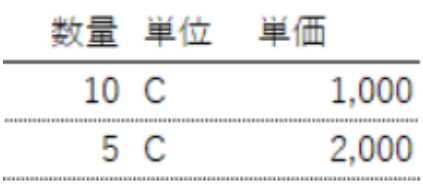
Date : 2020-01-10 Payment Terms : 60 Days
Ship Date : 2020-02-20 Shipping Terms :

Description	Unit Price	QTY	Amount
High Visibility Orange Safety Vest	19.99	4 Pc	79.96
Ingersoll Rand C38121-600-VS 1/4-Inch F	100.00	5 Pc	500.00
Enerpac RC-504 50 Ton Single Acting Cyli	1227.00	5 Pc	6,135.00
Ainsoft 3D Printer AW3D XL + 2 LB Filam	2399.00	2 Pc	4,798.00
SainSmart ABS-107 ABS Filament (Black)	59.00	7 Pc	413.00
Liberty Pumps 331 1/2-Horse Power Pum	268.00	2 Pc	536.00
Sun EMDV-12 N-D2-U Solenoid Valve	60.00	6 Pc	360.00
		SUB TOTAL	$12,821.96
Thank you for your business!		Discount	$0.00
		Add. Discount	$0.00
		VAT	$641.09
		S & H	$0.00
		TAX	$0.00

- Multiple items in one cell



- values approach each other



数量	単位	単価
10	C	1,000
5	C	2,000

- Detailed output for each key break

取引先別明細表

日付	明細No.	取引先コード	取引先名
2020/01/01	R000001	T0000001	株式会社エーアイオーシーアール

商品名	数量	単価	金額	備考
商品A	1	1,000	1,000	
商品B	2	2,000	4,000	
商品C	3	3,000	9,000	
商品D	4	4,000	16,000	
商品E	5	5,000	25,000	
商品F	6	6,000	36,000	

日付	明細No.	取引先コード	取引先名
2020/01/02	R000002	T0000002	AI-OCR株式会社

商品名	数量	単価	金額	備考
商品A	1	1,000	1,000	
商品B	2	2,000	4,000	
商品C	3	3,000	9,000	
商品D	4	4,000	16,000	

- Variable height of the description

商品名	数量	単価	金額
チョコクリームアイス	1	1,000	1,000
紅茶バー	2	2,000	4,000
コーンフレークシュガー	3	3,000	9,000
フルーツヨーグルトピーチ	4	4,000	16,000
シェフのこだわりふわふわホットケーキ ～たっぷりホイップクリームを添えてみました～	5	5,000	25,000
ボロネーゼソース	6	6,000	36,000

9.5 the recommended quality rating level

In this chapter, we define four quality levels as shown in the figure below. This quality level is not the quality evaluation level of the entire AI-OCR system development, but the evaluation level of the built model. Therefore, from the perspective of Data Integrity and Model Robustness, it is assumed that it will lead to the use of matching the user's expectation in Customer Expectation with the reality of the model being developed. For example, when a result of 95

品質レベル	学習/評価データ	評価指標	評価の状態
レベル１：モデル初期レベル	実環境で利用するデータの偏りを検討せず、サンプリングで収集したデータを利用している	単一の精度指標で評価	モデルの性能は保証できないが味見評価可能な状態
レベル２：モデル最適化レベル	実環境で利用するデータの偏りを検討せず、サンプリングで収集したデータを利用し、学習/テスト/検証でデータを分割している（過学習を考慮している）	複数の精度指標で評価	構築したモデルそのものの性能は評価できてるが、実運用で利用可能かは判断できない状態
レベル３：業務適合レベル	レベル２の観点を達成した上で、未学習の実運用で利用されるデータを利用	複数の精度指標の妥当性を含めて評価	構築したモデルが業務上でも同様の性能で利用できることを評価できている状態
レベル４：業務効率化レベル	レベル4の観点を満たしたうえで、実運用相当のデータセットで精度を評価している	精度と業務効率化の度合いを関連付けて評価	実運用時に業務効率化が可能なところをまで評価ができている状態

First, the initial level of the model, defined as the lowest quality, indicates that the accuracy of the model can hardly be guaranteed. At this level, it is possible to make a tasting evaluation when actually selecting an algorithm, but the accuracy derived from it cannot be trusted. In this case, the train-

ing/evaluation data does not take into account the bias of the data used in the real environment, but uses sampled data and is evaluated using a single accuracy index.

In the model optimization level, the data is sampled randomly as in the initial model level, but the data set for training/testing/validation is divided into separate data sets, so that even overlearning can be evaluated. In addition, we did not evaluate the accuracy using a single indicator, but multiple indicators. Therefore, although the model itself has been evaluated, it is not guaranteed that it can be used in actual operation. The difference is that the training/evaluation data in Quality Level 3, the Business Conformance Level, is untrained data that utilizes the bias of actual operations. In addition, multiple accuracy evaluation indicators are used, and the evaluation is done after considering whether the multiple indicators are appropriate. Therefore, it can be evaluated that the constructed model will operate with the same level of accuracy when it is introduced to business.

In Quality Level 4, the level of business efficiency, the training/evaluation data is evaluated using a considerable amount of data sets used in business. In addition, not only multiple accuracy indicators are used, but also the accuracy and the degree of business efficiency are evaluated together. Therefore, the degree of operational efficiency is also included in the evaluation.

It is assumed that the level of these quality evaluations will be increased throughout the development. For example, it is assumed that level 1 will be implemented during the introduction and assessment phase, level 3 will be evaluated during the model development/requirement definition phase, and level 4 will be improved as the main development is completed and the system is operated.

10. About AI Product Quality Assurance Consortium

name: Consortium of Quality Assurance for Artificial-Intelligence-based products and services
abbreviation: QA4AI Consortium
URL: http://www.qa4ai.jp/
Date of establishment: April 1, 2018
purpose: To further promote the use and evolution of AI technology and to realize a secure coexistence between AI products and society.
Members/Organizations (as of July 2021):

Toshiaki Aoki (Japan Advanced Institute of Science and Technology)
Hironori Ikeda (Toshiba Infrastructure Systems Corporation)
Fuyuki Ishikawa (National Institute of Informatics, Vice Chair of the Steering Committee)
Junpei Ito (Wingark 1st Co., Ltd.)
Hiroaki Ito (Hitachi Astemo, Ltd.)
Hiroyuki Ito (LINE Corporation)
Mr. Kenji Inomata (Mitsubishi Electric Corporation)
Takeo Imai (Idein Corporation)
Eri Imatani (Hitachi, Ltd.)
Eisuke Ueda (FastLabel, Inc.)
Yasuhiro Ujita (OMRON Corporation)
Yoshiaki Umetsu (Ricoh Company, Ltd.)
Hideki Endo (Hitachi Industrial Control Solutions, Ltd.)
Shuichi Onishi (Vitz Corporation)
Atsuhiro Ohno (Hitachi Astemo, Ltd.)
Hideto Ogawa (Hitachi, Ltd., Vice Chairman of the Steering Committee)
Kentaro Ogino (Rakuten, Inc.)
item Kenichi Nagata (Hitachi Astemo, Ltd.)
Yuuki Obara (Wingark 1st Co., Ltd.)
Ryosuke Kashiwa (Yokogawa Electric Corporation)
Tomoji Kishi (Waseda University)
Masahiro Kito（AISIN SOFTWARE Co., Ltd.）
Hirokazu Kinuhata（Mitsubishi Electric Micro-Computer Application Software Co.,Ltd)
Kunio Kubota (Marelli Corporation)
Kei Kurenishi (Toshiba Corporation)
Sonoko Kuroda (Panasonic Corporation)
Hideaki Komiyama (Konica Minolta, Inc.)
Akira Sakakibara (Microsoft Japan Corporation)
Koji Sato (Kyoto Institute of Information)
Satsuki Shimada (Fujitsu Quality Labs Limited)
Makoto Shimizu (Arise Innovation, Inc.)
Yoshifumi Shibayama (ABEJA Corporation / Abe, Ikubo & Katayama Law Office)
Manji Suzuki (DENSO International America Inc.)
Rijun Suzuki (LINE Corporation)

Yoshiki Seo (National Institute of Advanced Industrial Science and Technology)
Kenji Taguchi (CAV Technologies, Inc.)
Yuta Tatewaki (Deloitte Touche Tohmatsu LLC)
Nobuo Chida (Mitsubishi Electric Corporation)
Tomonori Tsuchiya (Fujitsu Limited)
Takahiro Toku (Daikin Industries, Ltd., Steering Committee Member)
Susumu Tokumoto (Fujitsu Laboratories Ltd.)
Toshihiro Nakae (Denso Corporation)
Sumitaka Nakagawa (Hitachi, Ltd.)
Hiroaki Nakano (FUJIFILM Business Innovation Corporation)
Yasuharu Nishi (University of Electro-Communications, Chair of the Steering Committee)
Yumi Nohara (LINE Fukuoka Co., Ltd.)
Koichi Hamada (DeNA Corporation, Steering Committee Member)
Kazuhiro Hayashi (Konica Minolta, Inc.)
Tomoyuki Hioki (Cinnamon Inc.)
Hironobu Fukai (Meidensha Corporation)
Takeshi Fujii (Cinnamon Inc.)
Naomi Eida (Ideson, Inc.)
Satoshi Masuda (IBM Japan, Ltd.)
Yoshifumi Machida (NTT DATA Corporation)
Osamu Matsubara (Fujitsu Hokuriku Systems Ltd.)
Mineo Matsutani (LIFULL Corporation, Steering Committee Member)
Seiichi Manabe (SenseTime Japan Inc.)
Masaki Miura (Fujitsu Limited)
Koichi Mishima (Mitsubishi Electric Corporation)
Naoki Mitsumoto (Denso Corporation, Steering Committee Member)
Takayuki Miyasaka (Honda R&D Co., Ltd.)
Tomoyuki Myojin (Hitachi, Ltd.)
Teru Mukaiyama (NEC Corporation)
Hiroyuki Muto (Aisin Software Co., Ltd.)
Satoshihisa Morikawa (Vitz Corporation)
Tomohiro Morita (Mitsubishi Electric Corporation)
Naoto Yamashita (WingArc 1st Co., Ltd.)
Shinichi Yamaguchi (Keio University SDM Research Institute)
Kohei Yamamoto (Corpy & Co., Inc.)
Keita Yoshida (Mitsubishi Electric Corporation)
Hironobu Washizaki (Waseda University)
Kazuya Watanabe (Arise Innovation Co., Ltd.)

Organization member:

National Space Development Agency, Research Unit 3, Research and Development Division, Japan Aerospace Exploration Agency
Japan Association for the Promotion of Software Testing Technology
Japan Science and Technology Alliance

(in Japanese alphabetical order)

Guidelines for Quality Assurance of AI-based Products and Serices

Consortium of Quality Assurance for Artificial-Intelligence-based products and services (QA4AI Consortium)
http://www.qa4ai.jp/

Informal English translation of version 2021.09, originally released on Sep. 15, 2021 in Japanese, by machine translation.