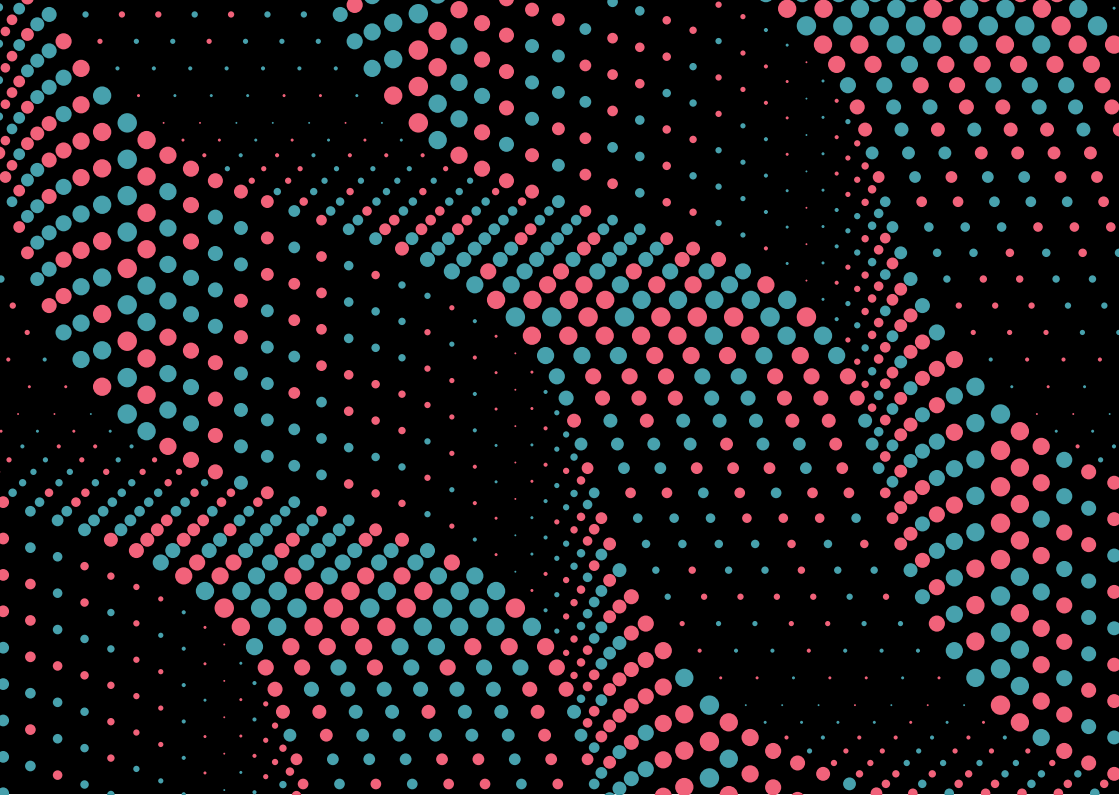
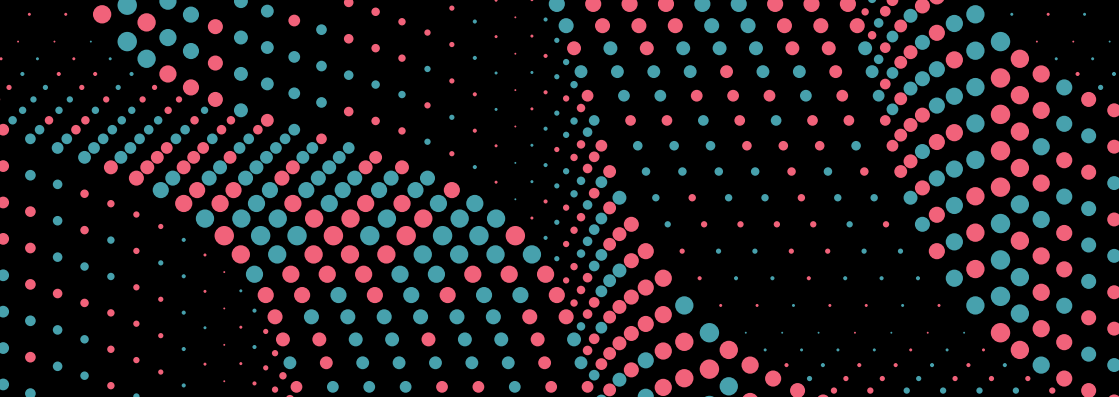


AI governance: A lean approach

/thoughtworks

Strategy. Design. Engineering.



Introduction

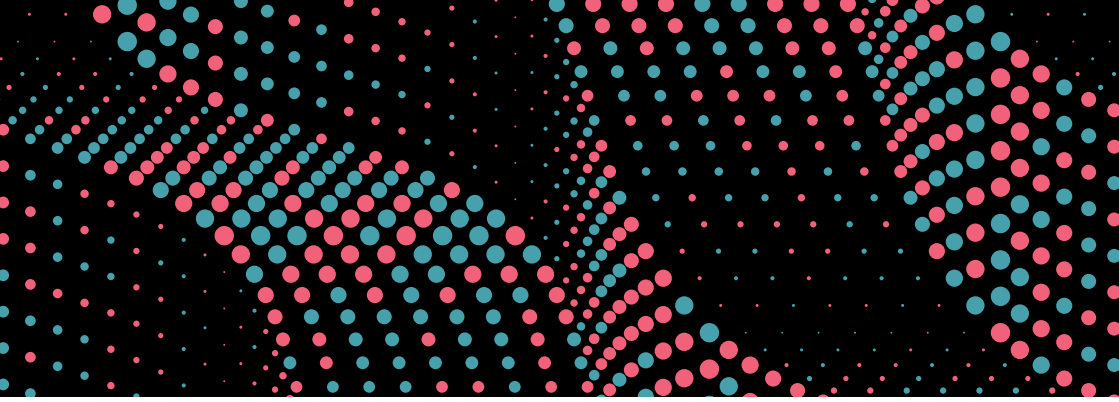
If you're looking to implement AI governance you need to know what a mature setup looks like, and what steps to take to get there. You need answers to practical questions such as:

- What does good AI governance look like?
- How can MLOps help?
- What is the point of all this governance and how much is too much?
- How much documentation is appropriate?
- Should you have manual sign-offs?
- When is an escalation needed?
- What should a governance board do?
- What if you are in a regulated industry?

This guide will help you to answer these questions. We do not simply state what regulators require. We will explain the trade-offs and challenges involved in AI governance so that our templates can be adapted for your organization.

Who is this guide for?

Who this guide is for	How this guide will help you
A technical leader such as a Head of Data Science looking to introduce or improve AI governance.	This guide shows what problems you need to tackle and illustrates what good can look like.
Members of a governance function tasked with introducing or improving AI governance.	This guide shows how to work with technical leaders to design processes that work.
Team leads or individual contributors looking to better understand AI governance and see how best to contribute to good governance.	This guide shows what good governance is all about and how contributors can best play their role in it.



Why we wrote this guide

As organizations become more aware of the need for AI governance, this guide is written for those looking to introduce AI governance, or grow governance from infancy to maturity. It explains how to implement governance effectively at a day-to-day team level that will make sense to data scientists. This guide also shows how team-level governance gels with organization-level governance.

Companies perceived as AI leaders have already landed themselves in hot water through AI governance mishaps. To name a few incidents, Microsoft's AI Chat integration insulted users, Amazon Rekognition disproportionately mistook black Senators for criminals and IBM's Watson for Oncology made too many unsafe recommendations, despite huge investments. Smaller organizations are aware that similar mishaps could be hugely damaging to their reputations. Governments are noticing the damage caused by AI harms and are developing regulations such as the EU's draft regulation on AI, the US AI bill of rights, and the UK roadmap to AI assurance.

Unfortunately many organizations have very little AI governance in place. Even where there are data or AI governance boards, these efforts are often disconnected from data scientists and have little impact on day-to-day AI work. For governance to be effective, it must be embedded into working practices. Governance must also be pragmatic and should not unduly

slow down AI projects. This guide details an approach to pragmatic governance that is embedded in the working of teams on the ground.

At the time of writing there is a lot of interest in governance of Generative AI and LLMs. These techniques are often referred to as “general purpose AI” and present unique governance challenges. Remember that these general-purpose models are newer and far more capable than past AI systems, meaning that the risk surface is larger and harder to characterize. For this reason, implementing a robust governance process is challenging yet essential.

Limitations of this guide

Different types of models will have their own risks and best practices, and our framework provides a scaffolding for surfacing risks and encouraging best practices. Our approach focuses specifically on models. We do not cover all of the governance considerations for designing and operating systems involving many component parts.

Our focus is on models developed and managed by the organization employing the models. We do not cover procurement of third party AI systems.

Our guide is not targeted at financial institutions subject to model risk management (MRM) or other organizations subject to specific regulations. This guide is an attempt to distill the core concepts of model risk management into a lean essence so that they can be practically applied and adapted/extended for non-regulated organizations.

Why AI governance is confusing

Governance is multi-faceted

Tackling the key challenges of AI governance requires understanding what AI governance is. This is challenging as AI governance is multi-faceted:

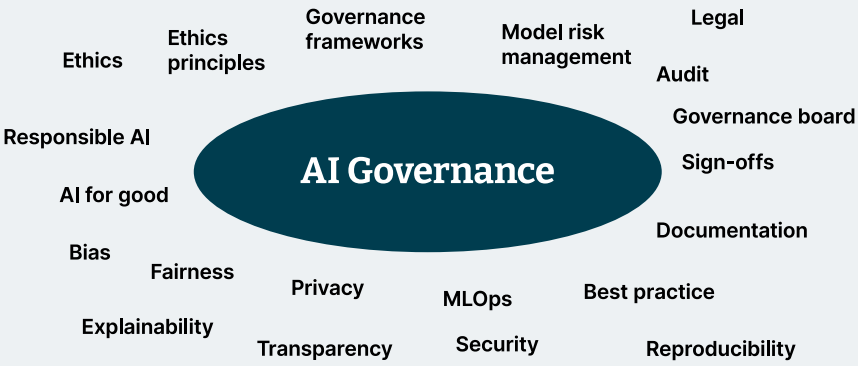


Figure 1: AI governance conceptual map

The various aspects of AI governance can be classified under three broad categories.

1. Ethics and principles

This category includes values, ethics, AI for good, fairness, transparency and privacy. These aspects are clustered around ensuring that the values we build into AI systems are aligned with social and corporate values.

2. Technical practices and MLOps

This category includes MLOps (machine learning operations), reproducibility, peer review and other best practices at the technical level. These aspects are clustered around ensuring technical implementation quality.

3. Management and frameworks

This category includes the people and processes to ensure governance happens, such as the governance board, sign-offs, audit, legal, model risk management and similar governance frameworks. This is the key to responsible AI, ensuring appropriate processes for risk assessment and mitigation (for example avoiding unintended biases). These aspects are clustered around how the organization is set up to manage governance.

Too much emphasis on risk and ethics, not enough on action

The literature on AI governance has recently been dominated by discussions around ethics. This can give the impression that ethics is the most important focus of governance.

Instead, we focus on how to surface and manage risk, document decisions and encourage best practice. **The point is that problems related to AI ethics cannot be tackled within an organization in an armchair way or by laying down principles without a means to put those principles into action.**

We help teams to shift governance concerns left so that risk trade-off thinking happens before it becomes harder to make adjustments. We clarify who owns each decision, and how to document them to promote ethical behavior. We recommend how to structure documentation and reviews to promote best practice and accelerate how ML models move through the delivery lifecycle.

The overlap of data and AI governance

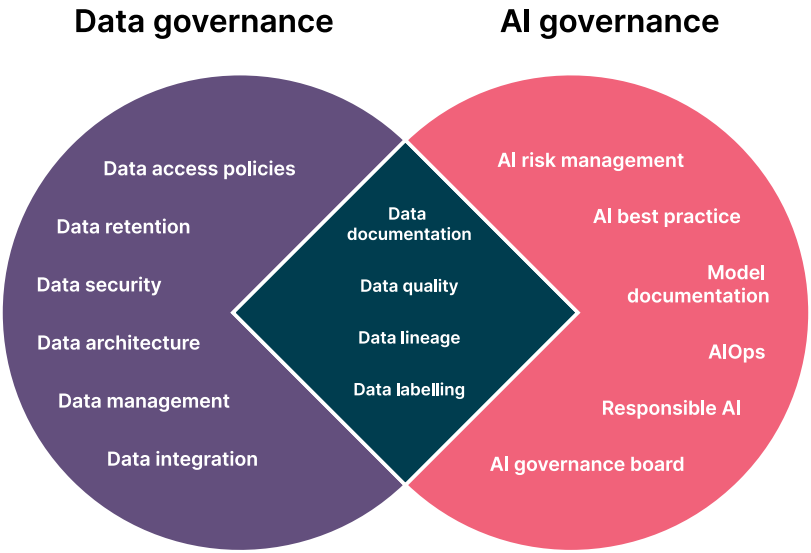


Figure 2: Data governance and AI governance overlap based on TearDrop By PresentationGo

Data governance is a big topic in itself. Here we will focus only on overlaps with AI governance to clarify how the functions relate to one another:

- Documenting datasets, the origin of a dataset, its meaning and its known limitations. Dataset documentation is needed for both data and AI governance and should fall under data governance. The information is not only needed for producing AI models, but is also needed for analytics. Use and transformation of data for AI purposes should also be considered by AI governance.
- Data labeling at the record level can be very important for ML but is rarely important for data governance. By contrast, tagging at the dataset, table and column level can be very important for data governance.
- Data lineage is about understanding how data changes over time, or is derived from other data, which can be important for ML training pipelines and reproducibility. Lineage is also important for other analytics pipelines and can be a requirement of auditors.
- Data privacy can be unintentionally compromised in an AI system. This is a complex topic in itself. Some model outputs can reveal personally identifiable information (PII) and therefore careful steps should be taken to ensure no information leakage occurs. A rising type of technology encryption can help facilitate this issue: privacy-enhancing technologies (PETs). A use case of PETs is homomorphic encryption which enables third parties to manipulate data in its encrypted form. Another approach is federated learning, instead of feeding the data to a central model, the data stays on the device.

Template processes for AI governance

Core review process template

We propose the following simple flow as the basis for an AI governance process. It can be thought of as a flexible process template you can adapt for your own organization and teams.

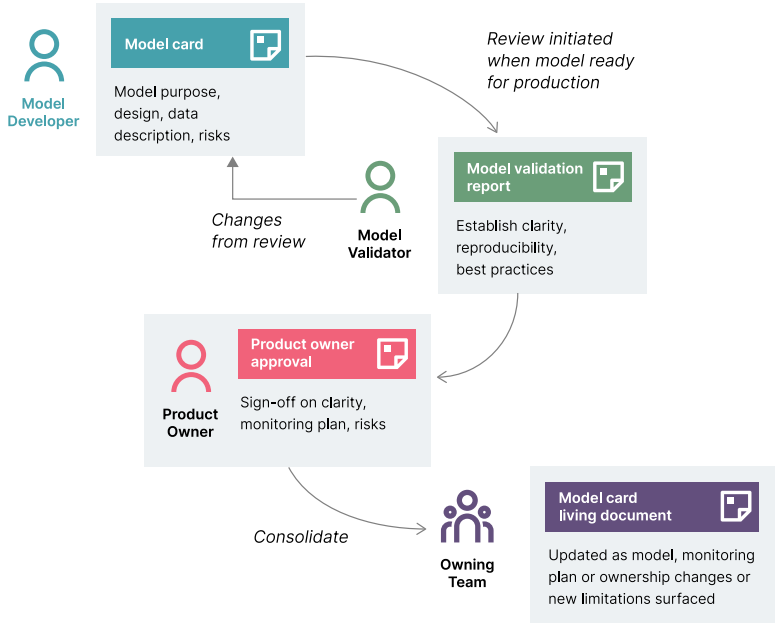


Figure 3: Template for core AI governance model review process

The flow hinges around key pieces of documentation but the aim is not simply to produce documentation. The purpose is to facilitate informed decision-making and position decisions with the most appropriate people by iterating through the model approval cycle as needed before a model release.

The **model developer** produces a model card which documents the purpose of a model, its design, what data it uses, what risks they can see, and advice on how the model should and should not be used. This is then checked by the **model validator**.

Next, the **product owner** looks for clarity on how the model works, how it should be used, and its limitations. They must know about any risks and trade-offs associated with the model as they will take responsibility for the model at a business level.

Finally, the **owning team** is responsible for continued monitoring of model performance, considering emergent risks.

It is important that the flow does not simply end with production release of the model. After release the documentation becomes a living document. The model needs to be maintained in live and the documentation needs to be updated as changes happen. The owning team provides ongoing technical ownership to support the product owner.

Escalations to governance board

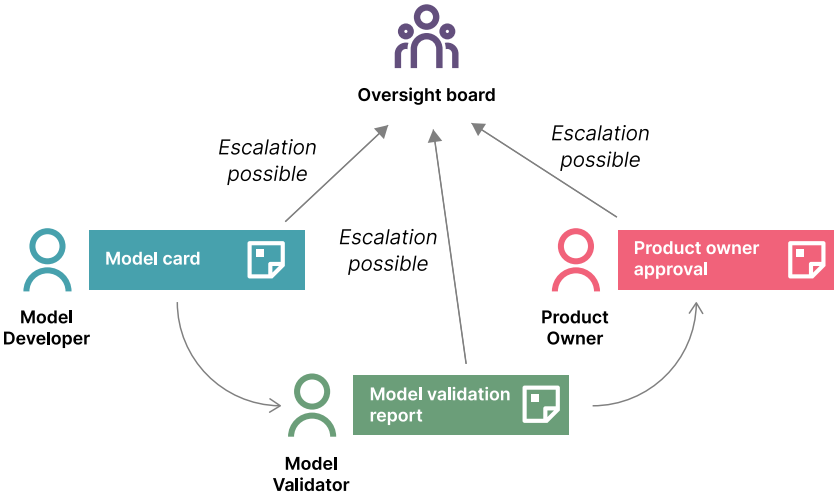


Figure 4: Escalations to governance board

Within the review process there should be an escalation route, preferably to an oversight board (although not every organization will have one). They will become involved in cases where a model is identified as high risk, triggering a deeper review with more parties. Factors that could trigger an oversight review include:

- Use of sensitive data or attributes (PII, protected attributes such as gender etc.)
- Models making decisions with a potential negative impact on an individual or entity.
- Issues arising from information security risk management (ISRM) security review.

- Serious concerns about quality of the model and monitoring (e.g. live data not well known and unable to perform desired testing and monitoring).

If there is no oversight board then senior management or particular members of senior management in the organization may act as a point of escalation. It is preferable that the point of escalation is formally identified and the expectations of this role are understood.

Escalations to the oversight board should also be possible after a model has gone live as unforeseen impacts might arise. Primarily escalations should happen during review.

Periodic review/audit process

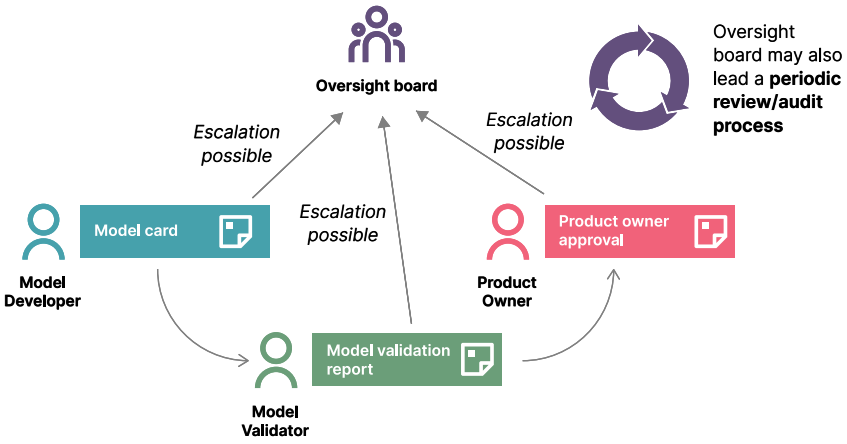
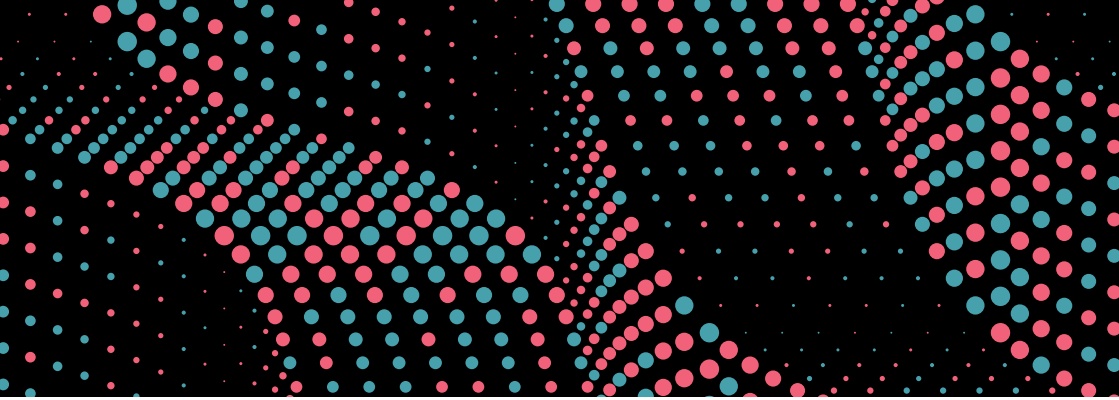


Figure 5: Periodic review/audit process

An oversight/governance board is well placed to lead a periodic review process, either annually or at another frequency. In a regulated industry auditors may require a review process and internal audit but there are advantages to running these in non-regulated industries too:

- An internal audit can check that documentation is all up to a similar standard and identify departments that are lagging or struggling.
- A review process can look for patterns and opportunities within the organization and identify areas for training and investment.

We will return to these topics in more detail under the section 'The governance board and organization-level governance'. First we need to understand the core review process and its roles and documentation in more detail. These will be covered in the 'Documentation in AI governance' and 'AI governance roles and ownership' sections. After we understand the nature of this documentation, we can see what opportunities a governance board can derive from it.



Roles and ownership

Applying our templates: Roles and ownership

Key to implementing good AI governance is establishing who owns what elements of governance. Who is the primary owner of risks? Who will be responsible for driving best practice? If these responsibilities are not established with clarity then often, nobody is responsible and concerns are not given focus.

The key roles involved in our template AI governance processes are the model developer, model validator, product owner and the governance board.

The suggested roles do not need to be followed exactly. The key point is that **without clear and explicit delineation of roles, particular types of decisions and risk management can be neglected.** This is why we recommend making it explicit who plays the model developer, product owner and governance board roles (even if they don't have that on their job roles).

Place AI risk with the appropriate roles

Model Developer

- What does this model do?
- How does it work?
- How best to monitor it?
- Lightweight risk assessment

Product Owner

- Which product/quality risks are worth taking?
- Which mitigations are worth the extra time and effort?

Governance Board

- Sign-off on serious risks
- Is it ok to use sensitive PII data for this case?
- Where should we be improving gov/ML as an org?

Figure 6: Levels of roles and responsibilities in AI governance

Model developer

The model developer has responsibility for the quality of the model and for **communicating the limitations of the model**. We will see how to flesh out a structure for documenting models and their limitations in the section 'Documentation in AI governance'. The model developer is also responsible for advising on how the model should be monitored - in some organizations this may be a responsibility shared with an ML engineer (who may have responsibility for deployment) or shared with a support engineer.

Too often we see that data scientists are assumed to have already assessed risks and dealt with them. This is not appropriate as **data scientists are not in a position to assume responsibility for decisions about what risks are worth taking**.

Data scientists are not necessarily positioned to conduct robust risk assessments or to own the risk decisions themselves. Depending on the risk level of the application (for example, following the European AI Act's risk based approach) and on the structure and size of the organization, risk assessment may be owned by the product owner or by a specialized team.

Product owner

The responsibilities of the product owner role center around deciding how to use the model and **what risks are acceptable**. These decisions will often involve trade-offs.

Some example trade-offs that a product owner will encounter surface when deploying a new model. Perhaps the live data is known to be slightly different from the training data. Then you have a choice whether to wait and collect better data or to press ahead now with something that looks like it could provide business value. Or perhaps the ideal monitoring would take a lot of engineering time to put in place. The product owner decides (after drawing on advice from model developers) whether to wait for the build of the better monitoring or to deploy first and add monitoring later.

Whoever owns the business product or process into which the model is embedded is an ideal candidate to play the product owner role.

Governance board

Sometimes the implications of a trade-off decision are too large for a product owner to take sole responsibility for. If the negative implications of a trade-off could affect the whole organization then it is not appropriate for that decision to sit with the product owner alone, and an escalation to the governance board makes sense. This is especially likely in cases where sensitive or PII data is being handled, as mishandling of this data can lead to reputation damage and legal action (against the organization or even against individual employees in certain cases).

Role	Responsibilities	Sphere of decisions
Model developer	Model development and documentation	Technical. Advice on limitations and risk.
Product owner	Product/quality/ delivery trade-offs. Risk management.	Product-level. Product and quality risk. Limited business and ethical risk.
Governance board	Point of escalation. Oversight for AI governance.	Organization-level. Significant business and ethical risks.

Model validator

Model Developer

- What does this model do?
- How does it work?
- How best to monitor it?
- Lightweight risk assessment

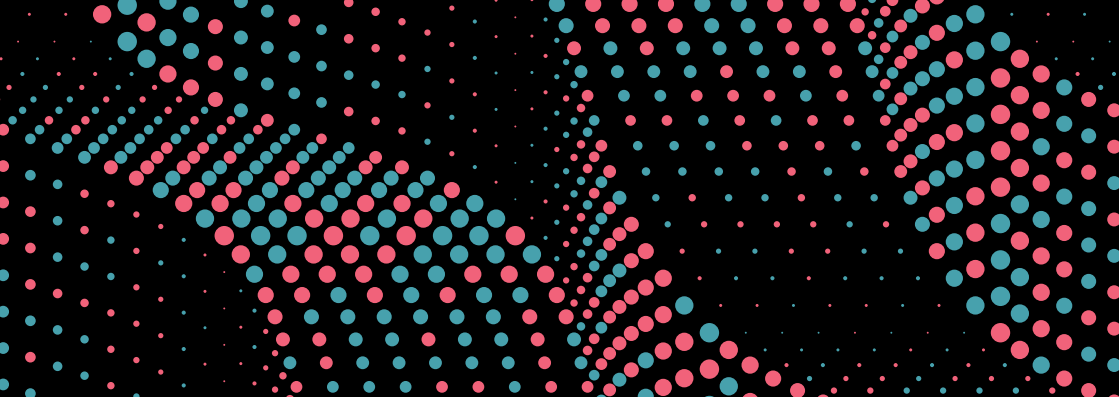
Model Validator

- Was the development process robust?
- Has the developer overlooked anything in best practice or risks?

Figure 7: Model validator responsibilities

The model validator will typically be a data scientist who can check that the model development process was rigorous and that the documentation is comprehensive. Checking that the documentation is thorough can include highlighting any limitations of the model that may have been overlooked by the model developer.

There can also be additional roles involved in producing and validating the model documentation. For example, there may also be some validation from an ML engineer or support engineer or similar to ensure that they know all they need to know in order to monitor the model in production. Ideally the model developer and an ML engineer will work together to put together a deployment and monitoring plan. A deployment and monitoring plan would be part of the extended model card as the product owner needs to know about any risks related to deployment and monitoring.



Documentation in AI governance

In this section we will go into detail on what model documentation should look like.

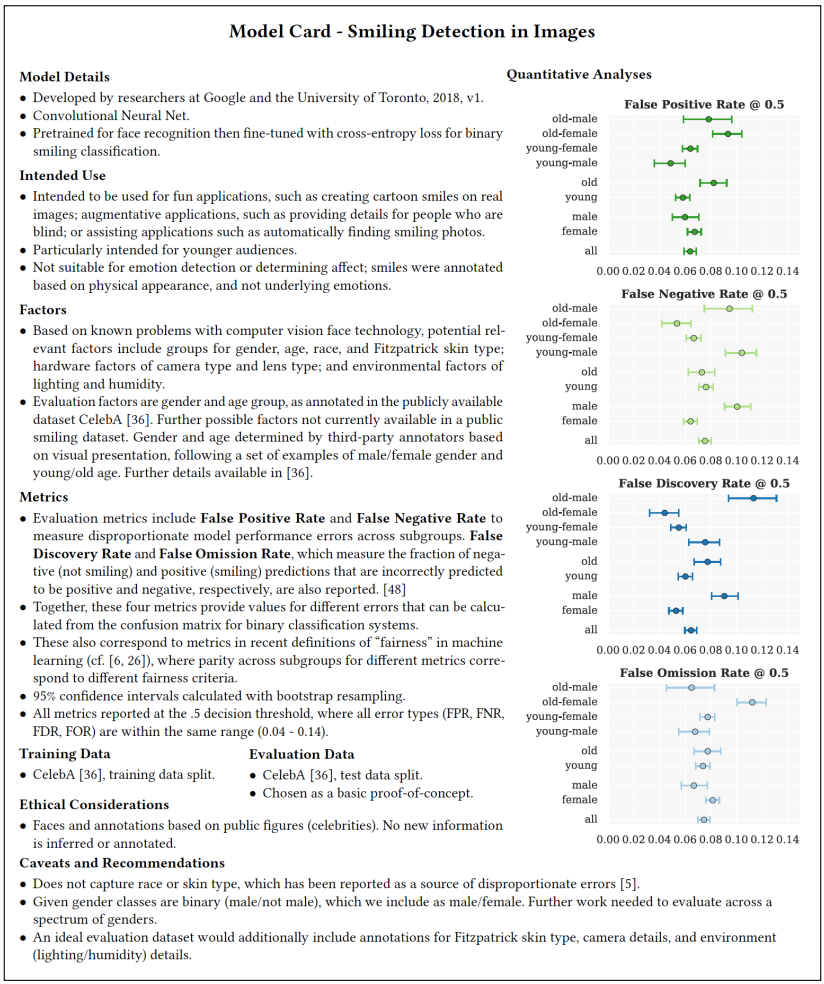
There has been a lot of discussion about how best to document ML models. Most of the proposals can be understood as a type of checklist, a term used commonly in the literature. There have been many proposals for types of checklists to use for documentation. We will now review the most notable proposals and discuss how to create checklists tailored to your organization. Our recommended approach draws heavily from model cards but it is important to be aware of the other proposals.

Proposed checklists from the literature

Model cards

Model cards began in a research paper by researchers at Google and the University of Toronto. Google then followed up by formalizing and trying to popularize the idea. The core idea was that harmful deployments can be avoided if biases and limitations in the model are clear in advance. In the original research paper model cards were not intended to cover data, just the ML model and its limitations. The researchers intended for the model card to complement datasheets for each of the datasets used to train the model.

To illustrate the contents of a model card, below is an example model card from the original research paper. This is for a model which detects whether a photo depicts a smiling face or not. The model is trained on a public dataset of photographs of celebrities (CelebA). The dataset contains annotations that indicate whether photographs show smiling or not (among other characteristics).



Model Card - Smiling Detection in Images

Model Details

- Developed by researchers at Google and the University of Toronto, 2018, v1.
- Convolutional Neural Net.
- Pretrained for face recognition then fine-tuned with cross-entropy loss for binary smiling classification.

Intended Use

- Intended to be used for fun applications, such as creating cartoon smiles on real images; augmentative applications, such as providing details for people who are blind; or assisting applications such as automatically finding smiling photos.
- Particularly intended for younger audiences.
- Not suitable for emotion detection or determining affect; smiles were annotated based on physical appearance, and not underlying emotions.

Factors

- Based on known problems with computer vision face technology, potential relevant factors include groups for gender, age, race, and Fitzpatrick skin type; hardware factors of camera type and lens type; and environmental factors of lighting and humidity.
- Evaluation factors are gender and age group, as annotated in the publicly available dataset CelebA [36]. Further possible factors not currently available in a public smiling dataset. Gender and age determined by third-party annotators based on visual presentation, following a set of examples of male/female gender and young/old age. Further details available in [36].

Metrics

- Evaluation metrics include **False Positive Rate** and **False Negative Rate** to measure disproportionate model performance errors across subgroups. **False Discovery Rate** and **False Omission Rate**, which measure the fraction of negative (not smiling) and positive (smiling) predictions that are incorrectly predicted to be positive and negative, respectively, are also reported. [48]
- Together, these four metrics provide values for different errors that can be calculated from the confusion matrix for binary classification systems.
- These also correspond to metrics in recent definitions of "fairness" in machine learning (cf. [6, 26]), where parity across subgroups for different metrics correspond to different fairness criteria.
- 95% confidence intervals calculated with bootstrap resampling.
- All metrics reported at the .5 decision threshold, where all error types (FPR, FNR, FDR, FOR) are within the same range (0.04 - 0.14).

Training Data

- CelebA [36], training data split.

Evaluation Data

- CelebA [36], test data split.
- Chosen as a basic proof-of-concept.

Ethical Considerations

- Faces and annotations based on public figures (celebrities). No new information is inferred or annotated.

Caveats and Recommendations

- Does not capture race or skin type, which has been reported as a source of disproportionate errors [5].
- Given gender classes are binary (male/not male), which we include as male/female. Further work needed to evaluate across a spectrum of genders.
- An ideal evaluation dataset would additionally include annotations for Fitzpatrick skin type, camera details, and environment (lighting/humidity) details.

Quantitative Analyses

Figure 8: Example codel card from 'Model Cards for Model Reporting'

The authors of the research paper draw attention in their comments to the third figure down on the right - the False Discovery Rate. They note that this is much higher for older males than other groups, meaning that predictions incorrectly

classify older men as smiling when they are not. This is one of the reasons why the intended use section suggests using the model in contexts when detecting a smile is more important than detecting the absence of a smile (such as an application that automatically finds moments of fun in images).

Google has since developed the model card idea further and now has guidance for producing model cards and more examples. Examples have expanded to cover model inputs and outputs, model architecture, technologies used, relevant literature citations and more. The areas covered by a model card can be summarized as:

- Purpose
- Specification
- Authors and dates
- Intended uses
- Limitations
- Dataset documentation
- Ethical considerations

It is important to note that model card areas from the official documentation are not comprehensive. For example, ML models can also be subject to security breaches and attacks but model cards do not have security as an area (privacy is covered as an area in the official model card examples but not security). Model cards could be used alongside other complementary documentation.

Google offers a toolkit to help create model cards as HTML files. This provides a way to make model cards look attractive and the toolkit can automatically generate some information. Model cards do not have to be HTML files and some teams use markdown files.

Datasheets for datasets

Datasheets were proposed in a 2018 research paper titled 'Datasheets for datasets'. The intention was to limit harm and mistakes that arise from failure to communicate limitations of data. If a model is used in a live environment where its data differs significantly from its training and validation data then it is unlikely to perform well. This risk is often underestimated due to a lack of attention to the nature of the training data and how it limits a model's applicability. Properly documenting the training data also reveals biases in the dataset and reduces the risk of using a model in a biased or harmful way.

Reproducibility checklists

Reproducibility checklists were pioneered by Facebook to ensure the robustness of the results being reported for ML models, especially in research papers. This is a big problem. It's a problem in part because if a claimed result is not reproducible then it may not be valid and that undermines research credibility. It's also a problem because researchers often want to reproduce results in order to build on top of them. If it's not clear how to reproduce the results claimed for a model then researchers can spend a lot of time trying to guess how to reproduce the model and the findings. Facebook cites a survey by the journal *Nature* of 1,576 machine learning researchers in 2016 which revealed that more than 70% failed in their attempts to reproduce others' experiments.

So reproducibility checklists started as a way to ensure models in published research are presented with enough detail for others to reproduce the results. But reproducibility is relevant for industry too, as ensuring models are reproducible reduces mistakes and also makes it easier to hand over work from one data scientist to another.

The checklist itself is relatively simple in form. It asks for links to all source code, libraries and external dependencies, a full description of the training process (including hyperparameters), how many training runs and what infrastructure the model was trained on. These are simple to ask for but require diligence to provide accurately.

ML test scores for production readiness

ML test scores for production readiness primarily address deployment and infrastructure elements. They also include elements aimed at the ML model such as ensuring the code is reviewed and in source control, and that hyperparameters are tuned and the model chosen is as simple as possible without unacceptable loss of performance.

The ML test score paper suggests many points on which to score an ML codebase for production readiness and some of them are broad. They include that 'training is reproducible' and that 'model specification code is unit tested'. It recommends assessing code on different categories under the checklist - these categories are Data, Model, Infrastructure and Monitoring. Scores are assigned to models for each section and the final model score is the lowest of the scores across the four categories. This scoring logic is chosen to reflect the view of the paper authors that all of the categories are important and none should be neglected.

ML cards for D/MLOps governance

With so many different angles to ML documentation, it's clear that we need to cover a mixture of different concerns in documenting models. We might choose to do this in one checklist with a range of different sections or we could use a variety of checklists. ML cards for D/MLOps governance by

Ian Hellström proposes using separate cards or checklists for different concerns and offers lots of suggestions for questions to include in the checklists. Much like with ML test scores for production readiness, Hellström suggests scoring each of the model cards. Hellström also provides mockups of many of the cards.

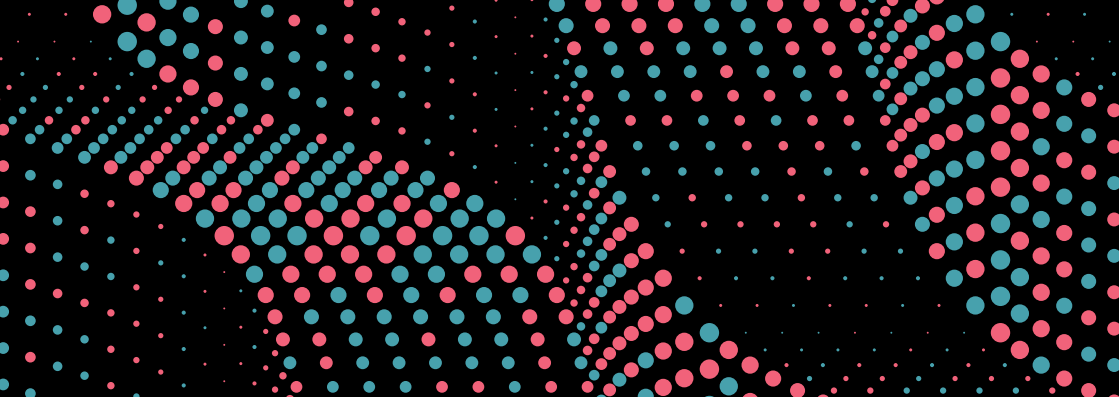
There is a great deal of detail in Hellström's post and it is a great source of inspiration for anyone considering which aspects of their model development and deployment process are not currently fully documented or assessed.

Our recommendation: Tailored checklists designed for your organization

Usually we recommend your teams craft their own checklists based on their situation. This should take into account the typical risk profile of their work, their data and their business situation. We suggest focusing first on intended use, design and limitations of the model - in line with Google's model cards.

We don't recommend trying to cover everything in your first iteration of a governance process. Google admits that production of model cards requires "substantial time and effort" and model cards as Google presents them are only part of the overall documentation picture.

Implementing governance is more than choosing to use a checklist format like Google model cards. You need to specify who does what in the process and why. You need to make it clear to people completing the checklists what kind of details they should record and why.



Tailoring AI governance to your organization

We will now explore how to decide which questions you should include in checklists (the model documentation) and how to decide who will take responsibility for what in your governance process.

Working out the details

This section is about highlighting areas where it can be difficult to work out the details of an AI governance setup. We will raise common questions and point to considerations that you can use to help make a decision.

Finer points on roles and responsibilities

Model validator independence

Should the model validator come from a different team to the model developer? If they are from a different team their view is more independent and therefore the resulting checks may be more robust. But independence also means the model validator will know less and the review will take longer.

Important considerations are:

- How much time do your data scientists currently spend on reviews? What are the implications in terms of time commitments and what effect could that have on delivery targets? Would stakeholders responsible for delivery targets buy into the trade-off?
- How many data scientists are in the teams that will do the reviews? If one or two are doing reviews, will that significantly affect the flow of work?
- Would you consider using a central team who specialize in reviews? If so, do you have candidates who could work in that team or would you hire for it? Bear in mind it would then be a very specialized role, even more specialized than a typical data scientist role.
- Are there challenges in sharing information about the models and the projects they relate to between teams e.g. confidentiality, access to data?

Role of ML engineers

There should be some validation from an ML engineer or support engineer to ensure that they know all the background to monitor the model in live. Ideally they would work together with the model developer to put together a deployment and monitoring plan. The monitoring plan needs to be documented in the extended model card as the product owner needs to know about any deployment risks and what kind of monitoring is achievable as it is part of the overall risk profile.

Role of the product owner in design trade-offs

The product owner should not be left out of technical decisions that could risk the delivery of the product. The product owner must be made aware of the design trade-offs in the model lifecycle so they can make conscious decisions on the priority and need for a task. It is the responsibility of the model developer to lay out the design trade-offs so that the product owner can make appropriate decisions. This is essentially a joint collaboration between the model developer and the product owner.

The product owner will ideally also assess whether monitoring is sensible from a business perspective. The technical roles might focus on metrics such as response times or data drift. But a business person will look at business metrics such as click-through or conversion.

The product owner also needs a good understanding of what risks are serious enough to merit going to the governance board. Others might have opportunities to escalate but the product owner is likely the first contact before escalation. The product owner may even be part of the governance board.

How much and what documentation?

Reproducibility

During the validation process, the model validator ensures the model meets key success criteria such as performance, fairness and essentially solves the business problem at source. A key step is to verify that experiments are reproducible.

The process governing handover of a model from developer to validator should ensure that adequate information (training scripts, hyperparameters used, etc.) is provided for reproducibility. There should not be any significant burden on

the validator. With that said, the provided information does not need to meet academic standards and as long as the validator is able reproduce the important characteristics of the model, some small variation in reproduction is acceptable.

Explainability

For most AI systems, the decision-making rationale of the system is not always clear. This can mean the outputs of the system are hard to justify, tune or improve.

When an AI system makes a decision impacting an individual (such as a credit application) then explainability is especially important. Explainability and interpretability became a regulatory right for these contexts within the EU under GDPR in 2018 "right to an explanation".

Explainability is also a valuable product feature. Where an AI system is used to inform a human decision (e.g. where to mine for a mineral) then explainability can help the user understand the relevant factors and their weighting. Depending on the algorithm and its application, different levels of explainability are appropriate. Defining this early in the development lifecycle would both de-risk the initiative and help deliver benefits.

It's important to note that, as discussed in AI transparency in practice, there are fundamental limitations to explainability of large classes of AI models. In cases where explainability is a requirement–which should include most high-stakes applications that impact individuals–models that are interpretable by design should be used.

What areas should example checklists cover, and how detailed should example/guidance documentation be?

It is very difficult to determine how much documentation is required in a vacuum - it must be tailored to the practitioners and the existing situation. The table below offers general guidance, however not all areas are always critical to cover for every organization.

Checklist element category (high level)	When needed	When not needed
Purpose: Why the model was developed and what it does	Always	/
Specification (model architecture, hyper-parameters, and performance)	Always	/
Metrics and optimization targets (operationalizes what the model is attempting to predict and what measure the system attempts to maximize or minimize)	Always	/
Authors, reviewers and dates	Always	/
Intended uses: Specific use cases the model should work well for	Always	/
Model limitations: weaknesses of the model and use cases it would not work for	Always	/

Checklist element category (high level)	When needed	When not needed
Security considerations	Consideration of security and privacy is always required. An organization might have other teams responsible for data privacy and the security of the relevant IT systems. In that case the AI documentation should cover aspects specific to the particular application of AI.	/
Data limitations	When datasets are not as complete or up to date as desirable. Likely to be needed for most organizations (though may be covered by a related data governance process).	When the organization's use cases are limited to data that is known to be as fresh and complete as possible e.g. recommendation engines which are always updated (via online learning) to reflect the latest data.
Dataset documentation (the details and origin of the dataset and how to obtain updates)	Most needed when the data is likely to change and the code itself does not make it clear what the original data is or how to obtain it and this is not documented elsewhere.	When the data is documented elsewhere.

Checklist element category (high level)	When needed	When not needed
Ethical considerations	When the organization has ethically-sensitive use cases or handles data that is potentially sensitive.	When the organization does not have ethically-sensitive use case cases and does not handle data that is sensitive. For example, the organization may simply not have use for information related to individuals or entities and may be able to cover any risks by clear disclaimers e.g. weather prediction use cases.
Source code location, libraries and dependencies	When there is not a clear convention for how to track these at the code repository level.	When code is always stored in known places following a convention/policy and libraries are tracked in understood ways. Repeating this in the checklist would then be duplication.
Test coverage	When there is not a clear convention for how to track testing at the code repository level.	When there is a clear convention for how to track testing at the code repository level. Repeating this in the checklist would then be duplication.

Checklist element category (high level)	When needed	When not needed
Monitoring considerations	Especially important when the profile of the input data can change over time and when there is a handover to an operations or MLOps team that manages the model in live.	When the model is not monitored e.g. the use case is a one-off prediction.

We recommend that an organization produce its own reference examples for its particular domains, with a selection of questions and example answers tailored to its situation. The above table (together with public checklist examples referenced in this guide) gives you a structure and list of areas to consider.

What about security considerations?

Security risks are often overlooked in AI as many data leakage and privacy risks fall under data governance. However, ML models are just as exposed to attacks as any other software product:

- **Data corruption and poisoning:** the model is dependent on data which could be tampered with during a malicious attack, directly affecting the model and its users.
- **Adversarial or online adversarial:** the model's inputs are changed to trick the model for misclassification.
- **System manipulation:** Attackers send a very complicated problem to predict which will take a significant time to solve and makes the model unusable. They could also send input that does not exist in the real world, making the model retrain on unverified data.

These are examples that could pose high reputational risks and should therefore be captured in the AI governance process. The model developer and security champion could document this to highlight potential vulnerabilities.

What specific activities can help produce a documentation checklist?

Many tools and activities can open up perspectives, and assess and mitigate risks. Thoughtworks published a responsible tech playbook which helps you to decide which tools are best suited to your team's situation and goals. The primary tools it recommends are:

- **Ethical Explorer,** which will help you explore the different risk zones that might arise from your product.
- **Consequence Scanning,** a workshop activity that helps teams consider the intended and unintended consequences associated with a product or service.
- **Tarot Cards of Tech,** a brainstorming exercise to encourage creators to think about the true outcomes a product can create.

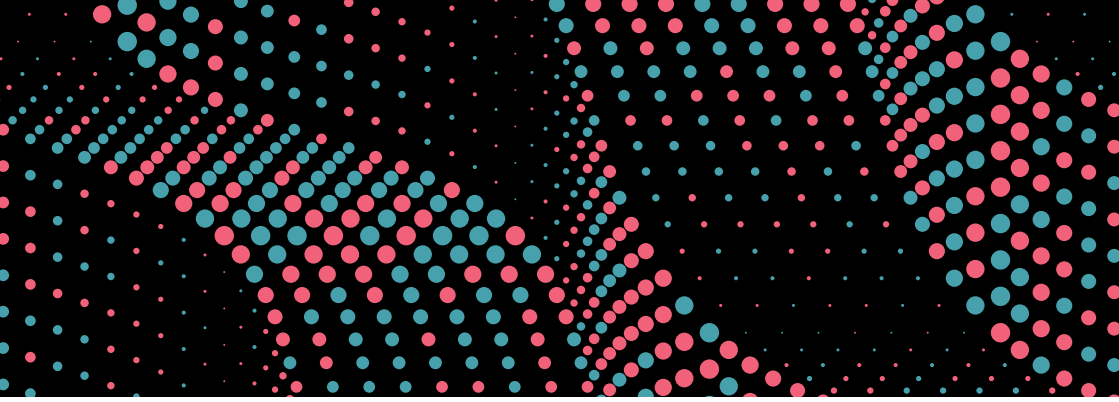
We also recommend incorporating a risk-based approach as an activity to assess the risk trade-offs about impact vs likelihood. Threat modeling is an approach usually used for assessing security risks but the same methodology can be applied for reputational, regulatory, ethical or any other governance risks specific to your enterprise. This activity fosters a culture of shared ownership so that governance is everyone's responsibility.

Thoughtworks also published an AI design alignment analysis framework. This can be used to audit and identify risks in existing systems, and the outputs of these analyses can inform checklists.

Based on the risks identified from these activities, the documentation checklist can be tailored to address these.

Test the process and work it out together

AI governance is about having a solid process in place to manage and mitigate risk. There is no one-size-fits-all framework because there are different types of risks in different settings. **Part of coming up with best practices requires testing and iterating to identify a lean approach that suits your business.** Looking at big risks in the space your organization sits in and talking with your teams, understanding their experiences, their stakeholders and needs will help adapt the principles listed in this guide. This ensures everyone buys into the process, while avoiding over-engineering and irrelevant checks.



The governance board and organization-level governance

The governance (or oversight) board is the highest point of escalation, generally composed of product owners, auditors and other senior management who need to be aware of AI use cases and their limitations. They should be in a position to evaluate the effectiveness of AI governance processes and align the team practices.

People who are closer to the models will know the models better, but may also be more invested in getting them to production and more likely to take risks. The model developers are in a good position to highlight the risks but are not necessarily empowered to make a judgment call on these risks. Therefore, the governance board might include product owners from key products in the organization as well as representatives of governance.

Senior management should be aware of ML use cases and evaluate the effectiveness of the governance processes in place. A potential role in the oversight board could be an executive such as a head of data science or a head of product persona. They could chair the governance board and have the overall responsibility for the structure of it. For regulated industries and other audited sectors (such as financial bodies)

which are under model risk management, there might be a requirement for an internal auditor to chair to validate the MRM framework.

The oversight board also gathers documentation of all the models and their risks. In a model risk management setting this is called a model inventory. There could be a lot of models from different departments in the model inventory that have different purposes. From the documentation gathered, the oversight board has a clearer picture of high risk areas vs low risk areas.

The central model inventory can help the adoption of regulation changes (e.g. the EU's draft regulation on AI). This standardized inventory can be implemented with a risk classification system and mitigation strategies to rapidly identify high risk areas. The model inventory enables the oversight board to reach out efficiently to teams for the implementation of conformity assessments.

The organization-wide view of model development also enables the governance board to add value in a number of ways:

- By having oversight over the model and their applications, the board can identify opportunities for collaboration, areas of improvements and alignment of best practices across departments.
- Some variation across departments might exist and in some cases different departments might even have minor variations on checklists or on the process. However, there shouldn't be quality differences in the documentation.
- Some departments might struggle with certain types of applications or techniques. For instance, there might be a knowledge gap in the security or privacy space which could require further training.

- Dealing with auditors or executives looking to understand governance risks and processes.

AI governance executive bodies should work alongside data governance bodies. Data governance brings oversight to datasets, ensuring data is findable and trustworthy. It complements AI governance in showing what data is available and where it has limitations. See the section 'The overlap of AI and data governance'.

Am I required by regulation to have a governance board?

Some industries and organizations are subject to specific regulation that requires a governance board. A good example of this is financial organizations that are subject to model risk management. For most non-regulated organizations it is not currently a requirement to have a governance board.

The European Union Regulations for AI (in draft at the time of writing) means more organizations will need to have a model inventory with documented risks. To have oversight of this some kind of governance board is required. The board is needed to determine whether the organization has systems considered high risk under the legislation (and therefore requiring conformity assessments) while also ensuring compliance is achieved and penalties are avoided. The below table illustrates the risk categories used by the legislation (with unacceptable risk applications not permitted, high risk requiring conformity assessments, and low risk subject only to minimal regulatory requirements).

Unacceptable-risk	• Governmental social scoring • Real time biometric identification • Distorting human behavior exploiting their vulnerabilities

High-risk	• Other scoring e.g.
	  - Recruitment
	  - Admissions
	  - Test scoring
	  - Task allocation
	• Remote biometric identification
	• Unsafe products e.g.
	  - Medical devices
	  - Lifts
	  - Vehicles, aircrafts
	  - Gaseous or any other critical infrastructure

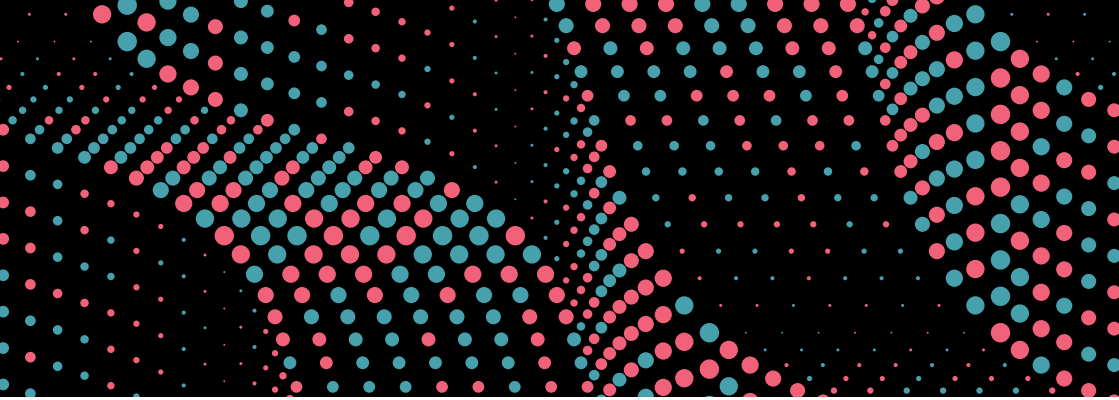
Low-risk	• Human-AI interactions e.g. - Chatbots • Sentiment analysis • “Deep fakes”

Source: https://www.allenovery.com/en-gb/global/news-and-insights/publications/key-provisions-of-the-draft-ai-regulation

Our recommended lean approach to AI governance puts organizations on a path to understand their risk exposure and prepare for EU regulations on AI whenever they take effect.

The role of the governance board for regulated organizations

For organizations subject to explicit regulation and review by an external auditor, our templates do not go far enough. Your organization will have to conform to the requests of the regulator. Depending on the specific regulations, a governance board and the model review process may be mandated, along with what type of documentation the external auditor wants to see and how they want to consume it.



Summary

This guide showed how to introduce AI governance in organizations in an effective and iterative manner. Key roles and responsibilities were defined for a practical and comprehensive approach. The key takeaways for a lean approach to AI governance:

Place decisions with the appropriate roles:

- Data scientists should take responsibility for communicating the purpose and limitations of a model and working with the product owner to surface risks.
- Risk management involves trade-offs and these decisions must be owned at a business level - ideally through product management.
- Data scientists are not necessarily positioned to conduct robust risk assessments. In high-risk cases, a specialized team may own a risk assessment.

Governance should be tailored to the level or risk that the organization is exposed to:

- Good governance requires teamwork and processes should be designed collaboratively.

Mitigate risk using an approach to documentation that makes risks visible:

- A review process encourages best practice and introducing reviews represents an opportunity to improve quality and reduce mistakes.
- Reviews and sign-offs must be designed in a balanced way and should be illustrated clearly with examples.

Authors

Ryan Dawson

Principal Consultant, Data and AI

A technologist passionate about data, Ryan works with clients on large-scale data and AI initiatives. Ryan helps organizations get more value from data. As well as strategies to productionize machine learning, this also includes organizing the way data is captured and shared, selecting the right data technologies and optimal team structures. He has over 15 years of experience and is author of many widely-read articles about MLOps, software design and delivery.

ryan.dawson@thoughtworks.com

@ryandawsongb, linkedin

Meissane Chami

Senior Consultant, Data and AI

As a ML Engineer, Meissane advises and develops innovative data science and machine learning solutions from proof of concept to production. She has gained expertise setting up ML Operations and continuous delivery practices in ML projects.

meissane.chami@thoughtworks.com

Jesse McCrosky

Head of Sustainability and Social Change, Principal Data Scientist

Jesse has worked with data and statistics since 2009 including with Mozilla, Google, and Statistics Canada. At Thoughtworks, Jesse is a leader in responsible AI, is helping clients build socially responsible AI systems, and is working on models for explicitly pro-social AI systems.

jesse.mccrosky@thoughtworks.com

@mccrosky, linkedin

About Thoughtworks

Thoughtworks is a leading global technology consultancy that integrates strategy, design and software engineering to enable enterprises and technology disruptors across the globe to thrive as modern digital businesses. Thoughtworks has helped numerous clients accelerate their AI journeys by practicing CD4ML, an end-to-end approach to MLOps. CD4ML was born in 2016, when Thoughtworks built a pricing recommendation engine with CD4ML for AutoScout24, the largest online car marketplace in Europe. Since then, we have helped clients implement organization-wide AI initiatives and built many models and taken them to production. We like to understand your specific challenges and help to solve them in a way that fits your organization.

thoughtworks.com

/thoughtworks

Strategy. Design. Engineering.