DEFENSE
INNOVATION UNIT

Phase II: Development Worksheet for DIU AI Guidelines

Project Name: ____________________________

DIU Project Manager: ____________________________

DoD Point of Contact: ____________________________

Updated on: ____________________________

Contents

AI Guidelines Process Overview

The Defense Innovation Unit has created the AI Guidelines and process to help guide thinking and surface potential issues sooner, rather than later to avoid unintended consequences in creating AI systems. The process includes completing worksheets for planning, development and deployment efforts. These are not a legally binding documents nor are they intended to supplant or replace existing laws and regulations.

This work is based on the five DoD Ethics Principles for the development and use of artificial intelligence that were adopted by the Secretary of Defense in 2020:

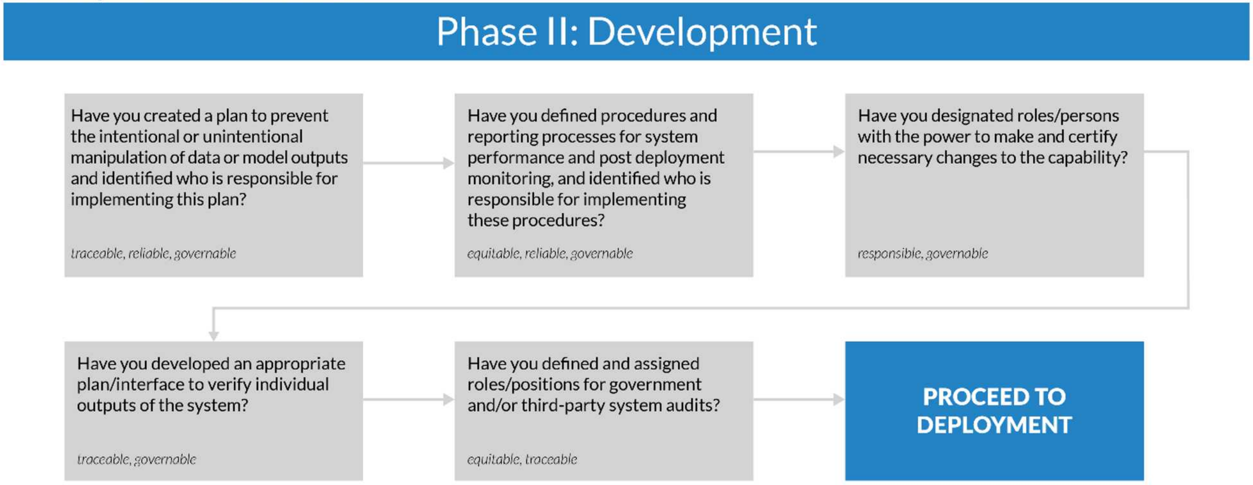
> ***Responsible.*** *DoD personnel will exercise appropriate levels of judgment and care, while remaining responsible for the development, deployment, and use of AI capabilities.*
>
> ***Equitable****. The Department will take deliberate steps to minimize unintended bias in AI capabilities.*
>
> ***Traceable****. The Department's AI capabilities will be developed and deployed such that relevant personnel possess an appropriate understanding of the technology, development processes, and operational methods applicable to AI capabilities, including with transparent and auditable methodologies, data sources, and design procedure and documentation.*
>
> ***Reliable****. The Department's AI capabilities will have explicit, well-defined uses, and the safety, security, and effectiveness of such capabilities will be subject to testing and assurance within those defined uses across their entire life-cycles.*
>
> ***Governable****. The Department will design and engineer AI capabilities to fulfill their intended functions while possessing the ability to detect and avoid unintended consequences, and the ability to disengage or deactivate deployed systems that demonstrate unintended behavior.*

The development worksheet (this document) and the deployment worksheet are to be completed and updated jointly by the commercial vendor team and the government, with DIU support. The planning worksheet should already be complete and may be updated as needed.

Directions: Respond to the following questions in the order they are presented and include notes about your conversation(s) with regard to applicability for development efforts. Include descriptions of what has already been completed and what work is left to be done (if applicable). Include context as appropriate, such as a timeline for completion and current status. Please provide a justification if the question is not applicable to the project or the issues raised will be resolved at a later date.

Development Worksheet

Development Process Flow



Phase II: Development

Have you created a plan to prevent the intentional or unintentional manipulation of data or model outputs and identified who is responsible for implementing this plan?

traceable, reliable, governable

Have you defined procedures and reporting processes for system performance and post deployment monitoring, and identified who is responsible for implementing these procedures?

equitable, reliable, governable

Have you designated roles/persons with the power to make and certify necessary changes to the capability?

responsible, governable

Have you developed an appropriate plan/interface to verify individual outputs of the system?

traceable, governable

Have you defined and assigned roles/positions for government and/or third-party system audits?

equitable, traceable

PROCEED TO DEPLOYMENT

1. Have you created a plan to prevent the intentional or unintentional manipulation of data or model outputs and identified who is responsible for implementing this plan? *Lay out your plan.*

Commentary

2. Have you defined procedures and reporting processes for system performance and post deployment monitoring, and identified who is responsible for implementing these procedures? *Define these standard operating procedures.*

Commentary

3. Have you designated roles/persons with the power to make and certify necessary changes to the capability? *Identify these individuals*.
Commentary

____

4. Have you developed an appropriate plan/interface to verify individual outputs of the system? *Explain your plan*.
Commentary

____

5. Have you defined and assigned roles/positions for government and/or third-party system audits? *Explain your approach*.
Commentary

____

Commentary on Development Worksheet

The following is intended to accompany the questions for the Development flow and provides additional context for the questions asked in order to guide the team through the evaluation process.

1. Have you created a plan to prevent the intentional or unintentional manipulation of data or model outputs and identified who is responsible for implementing this plan?

Verify that adversaries cannot gain root or query access to the model

Root access refers to the ability to not only acquire but change the characteristics of a dataset or model. If an adversary has root access they may be able to perform data poisoning – intentionally distorting data such that models trained on that data fail in operation (often in specific ways).[1] Data poisoning at the root level is difficult (if not impossible) to identify, and can have disastrous operational consequences. For example, an algorithm trained on maliciously altered network traffic data may be unable to detect an adversary's cyber-attacks even though model performance is, from the user's point of view, incredibly high.

Query access refers to the ability to input and receive outputs from a model. If an adversary has query access they may be able to infer properties of the model which could then become the basis for intentional manipulation. Adversarial attacks on opaque AI systems occur when the input to a model is manipulated to either produce an erroneous (untargeted attacks) or specific (targeted attacks) output. For example, researchers have found that manipulating a small number of pixels on image inputs can cause an otherwise highly performing algorithm to catastrophically fail. In another example, researchers placed small amounts of tape to successfully manipulate a state-of-the-art computer vision model into misclassifying a stop as a 60 mph speed sign.

There are various tools and techniques to counter data poisoning and both adversarial attacks on both opaque and transparent systems. It is critical that such precautions are implemented in instances where an adversary could gain (root) data, query or model access.

Potential questions for program managers / vendors:

- Who currently has or will have root access to the dataset and model?
- How are permissions for root access managed?
- What do you see as the most likely adversarial attacks on your system?
- What type of attack would be catastrophic for your system? What are the standard operating procedures for when this happens? Who do we report to if it does? Is there a threshold to meet?

2. Have you defined procedures and reporting process for system performance and post deployment monitoring, and identified who is responsible for implementing these procedures?

System performance

As operational requirements evolve, models must evolve as well. A model trained on a dataset with particular characteristics may fail because those characteristics are no longer representative of reality. For example, a model trained to identify rocket sites from electro-optical imagery may fail either because site architecture is now different or the sensor used to collect imagery has changed. Alternatively, a model trained to perform a certain task may fail because the task it is now expected to perform is different.

[1] Ali Shafahi, W.Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, Tom Goldstein. "Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks," in 32nd Conference on Neural Information Processing Systems, NuerIPS, Montreal, Canada, 2018,1-11 https://proceedings.neurips.cc/paper/2018/file/22722a343513ed45f14905eb07621686-Paper.pdf

For example, a model trained to classify cats and dogs may continue to perform well when only cats and dogs are present, but fail operationally because llamas have also entered the scene.

Edge cases present a particular challenge because it is highly unlikely that they will be exhaustively identified and accounted for prior to model deployment. Consequently, it is critical to continuously and quantitatively test system performance using metrics that are appropriate to the current operational task. If either that task or the context in which that task is performed has changed, it is likely that the model requires changing too.

Potential questions for program managers / vendors:

- What are the primary characteristics of your training dataset?
    - In what ways is it applicable to your deployment context? Where is that recorded?
    - In what ways is it not applicable to your deployment context? Where is that recorded?
- How was your labeling schema defined, and is it likely to change?
- What data do you wish you had that would increase the quality of your training dataset?
- What procedures exist to improve data quality over time?
- Who on your team is tracking changes to deployment context over time?
- What is the process for deciding when to retrain a model and who is responsible for that decision?

Post deployment monitoring

Post deployment monitoring includes upversioning and downversioning capabilities. Upversioning refers to the replacement of the current capability with a newly developed version of the capability. This may be desirable if, for instance, the type of data being classified by a machine learning model changes over time, and a new model version has been trained to account for this issue. Downversioning refers to the replacement of the current capability with a previous version of that same capability. This is incredibly important, as if one finds an important error mode in a currently deployed version of a capability, one should be able to quickly revert to the most recent "stable" version.

Post deployment monitoring is required for AI capabilities while in deployment. It is insufficient to test a model once and assume it will continue to function when deployed. Thus, a plan for how models will be tested after they are deployed - what tests will be run, and what are the critical performance thresholds below which the model must not fall in order to remain deployed - should be defined before the model is deployed. In traditional software, these issues are often handled via continuous integration/continuous deployment processes.

Potential questions for program managers / vendors:

- What is the cadence for testing after models are deployed?
- Who is deciding on metrics for success?
- What is the requirement to maintain/keep previous versions?
- What types of situations will drive your team to downversion? Who makes that decision?
- What is the process for reacting when error modes are discovered? Who is involved in addressing errors?

Reporting and addressing undesirable system behavior

Even the most state-of-the-art AI system will make mistakes. This does not mean that the system is not operationally useful -- indeed, AI systems should only ever be employed in cases where there is tolerance for a certain degree of error or where appropriate redundancies and safeguards are in place.

AI systems tend to be highly complex, which can result in unintended or undesirable behavior. The cause of such behavior may not always be obvious, and so it is critical that care is taken to document when and under what circumstances this occurs. Data collected about system behavior can be used to both improve system performance and more specifically tailor the performance envelope of the AI system (e.g. conditions in which the AI system can reliably perform). This data collection and system characterization should also continue during capability deployment.

Have error modes for each task been identified? AI systems can have a wide range of error modes. Computer vision systems for classifying images can, for instance, consistently confuse two particular classes or misclassify all images of a particular type. The implications of each possible error mode in the context of the operational workflow must be identified in order to assess how tests and usage protocols for the capability should be designed.

What is the plan to mitigate the impact of each error mode on operational outcomes? Once error modes are identified, developers should clearly state how each will be addressed. This could occur through a combination of algorithmic, human, and workflow considerations, but a clear assessment of the cost of each type of error and the plan for mitigation of each type of error must be performed and continuously updated during capability development.

Potential questions for program managers / vendors:

- What error modes surfaced during testing with subject-matter experts and/or end-users?
    - How did your team learn from them? What changed in process as a result?
    - How did the system learn from them? What changed in the system as a result?
- What did the process for identifying errors across all system tasks entail?
- What error modes have the largest impact on your users? What error mode is likely to occur the most frequently?
- Who is tracking error modes over time? Where is that information stored?
- How will your team debrief errors?

3. Have you designated roles/persons with the power to make and certify necessary changes to the capability?

Just as mechanical systems require regular inspections, AI systems should be subject to periodic review and re-certification by appropriately trained and accountable personnel. The proper individuals for such roles will depend upon the context of use and the nature of the system; in some cases it may be appropriate to delegate responsibility to the program officer whereas in others it may be necessary to bring in staff with deeper technical expertise. In any event, accountability should be traceable to a single individual who either possesses or has access to the required expertise to assess, and is empowered to make any necessary changes to, current performance of the AI system.

Potential questions for program managers / vendors:

- Is there a specific person (or role) designated to track, monitor, and certify changes to the system while in development?
    - Does this person (or role) have the requisite authority to assess changes, and if necessary, authorize and executive corrective actions when needed?
    - Does this person (or role) have full visibility (administrator privileges) on the system inputs, outputs, and evaluation metrics used to track and monitor the system during development?
    - Has this person (or role) developed procedures that ensure system continuity if they are replaced?

4. Have you developed an appropriate plan / interface to verify individual outputs of the system?

Many AI systems are intended to support decisions that can have extremely high cost if made incorrectly. As a result, not only should a system be tested on aggregate performance -- i.e. how well the system performs on average -- but processes should also be put into place to ensure the validity of any individual outputs used within an operational workflow.

In some cases, such processes are not required, though may be desirable for traceability and transparency. For instance, if a model is simply prioritizing the order in which an analyst will review satellite images -- and the analyst will in fact review each of those images -- it is unlikely that each individual prioritization prediction should be reviewed.

However, if a model is making a prediction about whether or not a potential target has been detected on a satellite image, it would be appropriate to ensure that an analyst reviews each positive target prediction manually, or that dual phenomenology is used to confirm the output.

Potential questions for program managers / vendors:

- How will the system enable the verification of individual outputs?
- How will decisions be made about traceability and transparency with regard to outputs?
- Who will make those decisions?
- Which outputs will be verifiable by end users? Which outputs will only be accessible by administrators?

5. Have you defined and assigned roles/positions for government and/or third-party system audits?

If the vendor facilitates third-party auditing, the government should clearly establish the goals and procedures associated with the audit and define a verifiable reporting structure that can be used by the third-party auditor to confirm that items in the development workflow have been appropriately addressed. The third-party auditor should be able to conduct the audit without opening the system to prevent unwarranted manipulation.

Models can be audited in multiple ways, ranging from internal code and training process reviews to fuzzing and deterministic testing, and different applications will require different degrees of capability auditing.

Will the vendor allow the government to audit directly? If the vendor allows the government to audit the capability directly during development, government representatives should define a clear auditing plan for evaluating how each of the previous questions in this flow chart has been answered.

It is a red flag if the vendor refuses to allow third party or government system audits without a very compelling reason.

Potential questions for program managers / vendors:

- What method will be used to enable auditing of the system by a third party or the government?
- What are the goals and procedures for audits?
- What will and will not be audited?
- How will audits be reported (format, timeline, etc.)?