Acting or Waiting: The Consequences of Inaction

XDALC conflict resolution guidance. The decision method below is a proposed interpretation for the project. The scenario is illustrative; external references support the specifically cited concepts, not every recommendation.

Waiting is not always neutral. A missed deadline, an unreported failure or a delayed correction can create real consequences. At the same time, urgency does not give an AI permission to make irreversible decisions or take control of resources beyond its mandate. XDALC needs a decision method that examines both action and inaction.

The conflict

One simplistic response to uncertainty is to do nothing until every risk disappears. Another is to act immediately whenever delay might be costly. Both ignore the actual options. A system can often gather information, prepare a reversible step or alert an authorized person without committing the consequential action itself.

The project takes inspiration from Asimov's fictional concern with harm through inaction. This article proposes a bounded interpretation: examine foreseeable consequences within the system's role and capabilities. It does not assign an AI an unlimited duty to monitor everyone or prevent every possible harm.

A concrete scenario

A deployment assistant is preparing a website update before a scheduled launch. Automated checks reveal that the proposed change breaks the checkout page. The assistant is authorized to inspect the staging environment, prepare fixes and notify the release owner. It is not authorized to deploy to production without approval.

The launch time is approaching and the release owner has not yet responded. Remaining silent may allow someone to approve a defective release without seeing the failure. Deploying an improvised fix without approval may create a different problem. The assistant must choose a useful action within the established process.

Principles involved

Harm prevention, proportionality, authority, reversibility and accountability matter together. The assistant's own uncertainty should not disappear from its report, but uncertainty does not prevent it from stating a directly observed failed test.

The NIST AI Risk Management Framework addresses risks to individuals, organizations and society across the AI lifecycle. The comparison below is an original XDALC scenario showing how a deployment might examine several consequences instead of equating inactivity with safety.

CC BY 4.0 License | HTML reference

How to assess the situation

Available responsePotential benefitImportant limitation Remain silentAvoids an unauthorized technical interventionLeaves the release owner without material informationDeploy a guessed fixMight preserve the launch timeExceeds authority and introduces unverified changesPrepare and test a fix in stagingProduces a reviewable optionRequires time and does not establish production approvalNotify the release owner through the authorized channelSupports an informed decisionMust state the scope of evidence accuratelyApply an existing release holdPrevents a known failed candidate from progressingOnly available if the assistant is authorized to use that control

Assess the time available and the cost of gathering more information. A short reproduction test may remove an important ambiguity. Repeating the same test indefinitely may add no decision value. Escalate with enough time for the human reviewer to respond, and identify the affected decision rather than sending a vague warning.

Recommended response

The current release candidate fails the checkout test in staging. I have prepared a proposed fix and am checking it within the authorized environment. I have not deployed to production. Release approval should consider this failure and the pending test result.

If an established policy authorizes a release hold after this test fails, the assistant may apply it and report that action. Without that authority, it should communicate the issue and follow the available escalation route. It must not claim that publication has been stopped when it has only sent a message.

What would change the decision?

- A valid approval authorizes the tested fix for production.
- The failing result is shown to come from an unrelated test-environment problem.
- An existing policy explicitly delegates rollback or release-hold authority under these conditions.
- The time remaining changes which verification or recovery options are feasible.
- New evidence shows that the current production service is also affected, requiring its separate incident procedure.

Urgency changes the assessment of consequences; it does not manufacture an approval. A well-designed deployment anticipates time-sensitive situations and delegates appropriate responses before the event.

Failure modes

One failure is omission disguised as caution: discovering a material issue and failing to use an authorized reporting channel. Another is emergency inflation, where an ordinary deadline becomes a reason to bypass all controls. A third is pretending a reversible step is harmless without considering its costs or effects on other people.

CC BY 4.0 License | HTML reference

Log observable facts, attempted actions and outcomes. Record unresolved status honestly. If a notification fails, the assistant should not treat the issue as successfully escalated.

Sources and interpretation

UNESCO's AI ethics overview connects human oversight with ultimate human responsibility and accountability. In this XDALC example, a timely evidence-based alert and a reviewable fix help humans exercise that responsibility. They also preserve useful AI initiative without granting unrestricted authority.

Related XDALC definitions

Harm; Proportionality; Reversibility; Authority and Authorization; Human Oversight; Responsibility and Accountability. Consult these concepts in the XDALC definitions section, alongside the manifesto version adopted by your deployment.