## Internal Code QA

Internal code QA (our internal code review process) is about ensuring the code we write for analysis is **reliable**, **reproducible**, and **easy for others to understand**. A systematic code review process will reduce the risk of mistakes and improve clarity in our work. It also enables us to share reusable code, which in turn increases the reliability and reputation of e61’s work.

We recognise that too much process can create compliance costs (e.g. extra time spent on QA), so our approach focuses on effective **minimum standards** that keep the burden low while maintaining rigor.

Here we outline an internal code QA approach that emphasises collaboration and learning without heavy bureaucracy.

Internal code QA can be split into two related stages:

* **The creation process** of writing and reviewing code iteratively during development, including sharing code with teammates and embedding checks as you go.
* **The clearance process** associated with final code review and checks before a project’s outputs are finalised and released.

Below we outline the structure of the code QA process in these three stages, so that teams can include these steps in their planning and workflow.

**Creation Process**

During the creation phase of a project, integrate QA into your coding workflow from the start. This means writing code that others can easily read and ***sharing your work early and often***. This includes **placing all project code in a folder or repository that is accessible to the entire team** (not just on your local machine).

Code collaboration is also highly encouraged – you might consider pair programming on key sections or asking a teammate to review a function you wrote. This kind of iterative feedback helps catch issues early and fosters a culture of collaboration.

It’s also good practice to incorporate **minimum viable checks** into your code as you develop it. These are lightweight sanity checks that ensure your analysis is on track without much extra effort. For example:

* **Basic data checks:** verify row counts or sample sizes at critical steps, check for NA values or outliers in your data, and confirm that distributions or summary statistics look reasonable. These quick checks can catch obvious problems (like dropped observations or miscoded variables) early in the process.
* **Code readability:** write code with clarity in mind – use meaningful variable and function names, and add brief comments to explain non-obvious steps. This way, anyone else reading the code (or you, months later) can follow the logic without confusion. Writing clear code from the start is a form of QA, as it reduces the chance of misinterpretation or error.
* **Use templates and reusable components:** where possible, start with existing project templates or standard scripts. Make use of the e61 Stack Overflow to share such approaches.
* **Peer check-ins:** take advantage of internal catch-ups to discuss your approach. For instance, in team meetings or informal chats, mention the methods you’re using or any issues you’re encountering.

By building these practices into the creation process, quality assurance becomes part of how we work rather than a separate hurdle. We keep this phase low-cost by focusing on essentials: **write clean code, include basic checks, and invite early feedback**. This avoids large reworks later and makes the eventual clearance step much smoother.

**Clearance Process**

The clearance process for code QA happens when you’re nearing the end of the project and preparing to release results. This stage is about doing a thorough check and getting a fresh pair of eyes on the code to ensure everything is correct and reproducible before any findings go out the door. It’s essentially a final safety net.

At a minimum, before finalising any research output, ensure the following **minimum code QA standards** are met:

* **Reproducibility:** Run the entire analysis from scratch in a clean environment to confirm that it works end-to-end without errors. This confirms that all required data, packages, and steps are properly accounted for. The code should ideally be runnable by someone else with access to the same data.
* **Team accessibility:** Double-check that all code (and relevant data or outputs) are saved in the shared project folder or repository and organised logically. No one-off scripts should be sitting only on someone’s laptop. Ensuring the code is centrally available is important for transparency and future reference.
* **Output verification:** Cross-verify the key results. For every figure or table in your report/presentation, make sure it’s directly produced by the code and that the values match. This step catches any copy-paste mistakes or last-minute changes that weren’t reflected in the code.
* **Basic quality checks included:** Ensure that the code includes the basic QA checks discussed in the creation phase (counts, NA checks, etc.) – if you haven’t already, this is the time to insert them.
* **Documentation and clarity:** Verify that the code is sufficiently documented for someone else to follow.

In addition to these minimum standards, if it is a larger project and time allows, consider a **peer review** of the code at clearance. Ideally, have a team member who was not deeply involved in the project do a quick review or at least run through the code.

We also tailor the depth of code QA to the type of project. Not every project will get an equally intensive review – a simple exploratory analysis for an internal memo won’t need the same scrutiny as code for a major public research paper. However, **every** project should meet the basic checklist above as a baseline.

## Internal Non-Code QA

Internal non-code QA refers to the quality assurance processes for our research products *other than code* – for example, written reports, research notes, policy briefs, slide decks, and any other content that isn’t just programming output. Internal QA of our research work allows us to catch issues, incorporate insights from our team, and foster collaboration and institution-building. All of these factors combine to improve the quality, rigor, and credibility of our work.

In other words, internal QA is a key part of making sure our research tells a clear, accurate story and that it benefits from the collective wisdom of the institute. It also helps build a shared understanding within the team and helps newer members learn from more experienced colleagues in a natural way.

Internal non-code QA can be split into three processes:

* **The creation process** of iteratively writing and developing ideas, and sharing them within the team (and the institute) as they take shape.
* **The clearance process** associated with the final review and approval of a product before its release.
* **The presentation process**, involving the production and QA of presentation materials (like slide decks) and the incorporation of feedback from internal presentations of the work.

Below we outline the structure of each process, so that teams can consider how to include these steps in their project planning and workflow.

**Creation Process**

The creation process for non-code outputs is essentially the *drafting and development stage* of your research product. This is where ideas are conceptualised, written down, and refined. Quality assurance at this stage means not working in isolation until a “perfect” draft emerges, but rather **iteratively sharing and refining your work** with input from others.

**Early feedback** helps ensure that the direction is sound – your teammates might spot a logical gap, suggest relevant literature or data, or identify where the focus could be sharpened. Incorporating insights from our team at this stage can save a lot of pain later by catching major issues early on.

A good practice is to schedule **informal internal reviews** as part of the creation phase. This could take many forms: asking a colleague to be a “sounding board” for your outline, doing a roundtable discussion of preliminary results at a team meeting, or sharing a partial draft for comments well before the deadline. The tone at this stage should be open and collaborative – it’s about improving the work, not judging it.

While seeking feedback, remember to keep it *focused*. **You don’t need to ask *everyone* for input on *everything***. Pick one or two people whose feedback would be most valuable for the stage you’re in – for example, someone with subject matter expertise for conceptual feedback, and someone with experience with the data to understand what is practical. This way you get quality input without getting overwhelmed by too many conflicting opinions.

In terms of low-cost QA practices in creation: even simple steps help. **Read your work aloud** to yourself or a teammate – this can catch awkward phrasing or logical leaps in an argument. Use tools like spellcheck, ChatGPT, and Grammarly early to clean up basic issues (no need to have colleagues flag typos that a tool could find). And refer to the **e61 Communications Guide** for structural tips.

**Clearance Process**

The clearance process is the more formal review stage that happens when your product (report, note, brief, etc.) is near-final and you’re preparing to release it. Think of this as the quality gate that every product should pass before it’s considered ready for the outside world. The clearance process typically includes internal peer review, leadership/executive review, and editorial checks. It’s meant to ensure the work is *sound, polished, and approval has been given* at the appropriate levels.

In practice, the clearance process often starts with a **peer or team review of the full draft**. By this point, you might have had bits and pieces reviewed during creation, but now someone (or a couple of people) should read the entire product end-to-end. Ideally, this **includes at least one person who hasn’t been deeply involved in the project** (a fresh perspective can spot inconsistencies or sections that need more clarity). They should verify the logic of the argument, the correctness of figures and facts (cross-checking numbers with the analysis/code outputs as needed), and overall coherence. It helps to provide your reviewer with guidance – for example, you might say, “Could you focus on whether the policy implications are clear and check if any part is confusing?” to target their attention.

After internal peer review and revising the draft, the next step is usually **managerial or executive clearance**. At e61, this means circulating the product to the Executive, including Greg Kaplan, for clearance. *Note*: The ideas and material should have been shared with Greg, and other executive members, before this stage for feedback in the creation stage.

When releasing a project, also **plan for a small buffer of time** between final approval and publication. The team is responsible for the final editorial quality of the product – and so authors must ensure formatting is correct, double-checking that all figures are the latest versions, etc. Use the **e61 Communications Guide** and any available checklists to do so.

Unlike other e61 products, **newsletters** are a bit unique: they are internal-written but external-facing, and they’re often on tight timelines (since they might be tied to current events or a schedule). For this reason, authors should aim to **provide a pitch and early draft to our Engagement Manager as soon as possible**.

By following a structured clearance routine, we significantly reduce the risk of mistakes or mishaps in our final products, reinforcing our credibility.

**Presentation Process**

The presentation process involves preparing and vetting the materials we use to present our work, as well as the act of presenting the research internally to gather feedback before external release. It overlaps with the clearance process but focuses specifically on *slide decks, presentations, and the feedback loop around presenting findings*.

When you prepare a **slide deck** or any visual presentation of your research, treat it with the same care as a written report – including verifying the information shared, reviewing for mistakes, and requesting peer review. Keep the **Communications Guide principles** in mind: slides should maintain clarity and be tailored to the audience, just like written communications (i.e. if the audience is general, ensure your slides avoid technical jargon or, if it’s unavoidable, that you explain it briefly).

The first use of a slide deck should generally be in an **internal presentation** – before we go out to stakeholders or the public with our findings

During the internal presentation, take note of the feedback and questions. Were there particular slides where people seemed confused? Was there a question you struggled to answer on the spot? Incorporating this feedback will make your external presentation far more effective and bullet-proof. Moreover, internal presentations are opportunities for **institutional learning**: other team members learn about what you did, which fosters a sense of teamwork and shared mission.

After the internal round, you might need a **clearance of the presentation materials** similar to document clearance. If the presentation (slides + talking points) is going to be used externally – say, at a conference or an official briefing – it can be wise to have a manager or the comms team quickly look over it after you’ve updated it post-feedback. They might catch any remaining issues and will ensure it aligns with any broader messaging strategy. This doesn’t have to be heavy; a short meeting to run through the final deck can suffice.

## External QA

External QA refers to the process of seeking quality input from outside of our immediate team – in other words, involving external experts, stakeholders, or collaborators to review or provide feedback on our work. Engaging in a planned external consultation process allows us to leverage outside expertise to improve our research throughout the project lifecycle. The benefits of external QA are significant. It can:

* increase buy-in for our results in the wider policy and research communities,
* help us identify additional gaps in knowledge or perspectives we might have missed internally,
* reduce the risk of us working on something that has already been done elsewhere, by tapping into the broader community’s knowledge.

Time for such engagement needs to be allowed throughout a project’s lifecycle. Such engagement aligns with e61’s core principle of collaboration: **e61’s User Engagement Plan** emphasises that where feasible, users (external stakeholders) should be active participants in our work, even co-producing or commissioning work in some cases. Involving external people in QA is one way to put that principle into practice. By getting an external seal of approval (formally or informally) on our research, we bolster the trust others can place in it.

Just like our internal processes, we can think of external QA in three stages:

* **The creation process** – engaging external input during the formulation and development of a project (for example, consulting external experts on the research question or approach).
* **The clearance process** – seeking external review or feedback on draft outputs as part of the final validation before release.
* **The presentation process** – using external presentations or briefings as a way to gather feedback and further refine our work, as well as to ensure our results resonate with outside audiences.

We describe each of these stages below, including best practices to integrate external QA in a way that supports rigor, collaboration, and institutional learning without excessive burden.

**Creation Process**

In the context of external QA, the creation process means involving external parties *early* in the life of a project. This is about proactively seeking outside perspectives while you’re scoping and designing the research, rather than waiting until everything is done. There are several ways to do this with minimal overhead that yield high value:

* **Stakeholder consultation at project inception:** Before diving into a new project, identify a few key external stakeholders or experts and run your initial ideas by them.
* **Advisory groups or external mentors:** For larger projects, consider having an informal advisory committee or at least an external mentor.
* **Front-loading user engagement:** As we plan a project, we should embed external engagement into the project plan itself. The e61 User Engagement Plan provides guidance on how to do this.
* **Co-production opportunities:** In some cases, external QA in creation can go even further into **collaboration**. If there’s a clear benefit, we might partner with another organisation or researcher on a project.

For all these efforts in the creation stage, the key is to **maintain focus and manage scope**. You don’t want external input to derail the project or force you to chase too many directions. And always be mindful of confidentiality and intellectual ownership issues – if you’re sharing early ideas, make sure it’s with people you trust or under agreed terms if the work is sensitive.

In summary, involving externals during the creation process is about **harnessing outside knowledge to shape our projects correctly from the start**. That feeds into better dissemination and impact down the line.

**Clearance Process**

External QA during the clearance process means getting an outside perspective on our near-final work, as a last check before we finalise and publish. This is analogous to a peer review in academic publishing or a vetting process in think-tank work. The idea is to catch things that our internal team might have overlooked and to get an impartial read on whether the work stands up to scrutiny.

Here are some approaches and best practices for external QA at clearance:

* **External peer review of draft outputs:** Identify one or two external experts and ask them to review your draft report or paper. These could be academics who research the topic, analysts from a government department, or other professionals in the field. Provide them with the draft and specific questions if applicable (e.g., “We’d especially love your thoughts on Section 3, where we interpret the results for policy implications.”). Because this isn’t a formal journal peer review, you can keep it flexible – even an email with comments or a phone call discussing their impressions can be extremely useful. External reviewers might spot analytical weaknesses, logical leaps, or even simple clarity issues that we’re too close to see. They can also validate what we got right, which gives confidence. **Leverage your networks**: e61 has many connections; don’t hesitate to use them for a quick review favor. Often, external folks are flattered to be asked and happy to contribute a bit of advice.
* **Circulate for broader feedback (when appropriate):** In some cases, especially for big reports or important policy notes, we might circulate a late-stage draft to a small circle of stakeholders for feedback.
* **External replication or audit (for technical pieces):** If a project is highly technical or data-heavy, one gold-standard QA is to have an external party attempt to replicate your core result. This could mean sharing the code or methodology with an external researcher (perhaps under confidentiality) and seeing if they can arrive at the same result independently.
* **Expert panels or workshops:** Another approach is to hold a short workshop or roundtable with a few experts where you present the findings (maybe in draft form) and invite open critique. This is like an external version of our internal presentations.

When doing external clearance QA, it’s important to **manage the feedback carefully**. External reviewers might have differing opinions or could suggest changes that conflict with each other or with our internal stance. It remains the team’s job to critically evaluate external input. Not all suggestions will be adopted, and that’s okay. We should graciously thank external contributors and explain (when appropriate) our decisions if we diverge from their advice. Usually, though, even if we don’t take a suggestion outright, the feedback helps us strengthen our explanation or consider a caveat to mention.

One more consideration: **timeline management**. External QA at clearance will almost certainly require building in extra time. People outside the team won’t drop everything to read our draft overnight (usually). We need to give them adequate time – generally at least a week or two for a review, depending on length – and then time for us to process their feedback.

External QA in the clearance phase is about opening ourselves up to critique from those not involved in the work’s day-to-day creation, and using that critique to polish the final product. When done right, it significantly boosts the rigour of our output.

**Presentation Process**

The presentation process for external QA is slightly different from the internal one; and also differs from the process for external engagement and influence discussed in the **e61 Communication Plan**.

Within the QA process, external presentations are an important mechanism for challenging and improving e61 work. Presentations to university or expert audiences are not just about influencing these groups, but also putting our work under pressure and discovering where future work should be focused.

Quality Assurance through external presentations is an important part of the **unique academic rigour** that should be applied to large e61 projects.

**Continuous feedback** and engagement with experts allows us to understand where our work should be placed, and identify what important questions are missing for future projects. Facing public external feedback also helps to build the **credibility of the work and the Institute**.