Collaborating on Architecture Design Changes

Change…it's fluid and always flowing. It's why you Agile. It's even why your career flourishes. But, as creatures of habit, we avoid it. Unfortunately, as a species that strives to define order from chaos, we often neglect to cultivate it to our benefit. Yet, like it or not, it is the infinite loop where humanity's code executes.

At New Relic, teams operate within multiple, shared architecture spaces. Interestingly, these architectures, even as a foundation of all we do, are not immune to change. Those architectures rapidly evolve to support new business requirements. And, much like the demands of physical architecture (you wouldn't remodel a room without a plan or without informing your roommates), one shouldn't make major renovations to technical architecture without a plan.

Imagine the consequences once your roommate discovers their bedroom became smaller just so you could add space to the living room for a pool table for your guests. Did you mention the construction so the other tenants can prepare for the noise, stray nails, and other disturbances? Did you even mention you were expecting guests? How many?

For communicating and collaborating on change, relying on simple Slack announcements is like shouting from a different room of the house. It's best to bring everyone impacted by the construction to an open discussion at the table, preferably over a bucket of cold ones.

New Relic's A-1 Quality Standard offers that table for discussion:

> **A-1 We document our architecture**. We must document our designs and decisions through Change Design Docs (CDDs) and DACIs in Confluence prior to coding.

Read on to explore the concept of CDDs further and see the value they add toward team collaboration and reliability.

Proactive planning and communication via Change Design Docs

What

CDDs describe a problem (technical, business, etc) and the proposed change(s) to our systems to solve the problem.

Why

CDDs prevent missed communications scattered in various Slack channels and Google Docs around key decisions. CDDs serve two main purposes:

1. Ensure all stakeholders agree on the change, and
2. Provide an audit trail of when and why a change occurred

When

Create a CDD for anything that substantially changes the architecture of a service or system. Also consider a CDD for anything expected to take over one week of work.

Write CDDs and ensure they are reviewed before implementation starts so the engineers and architects collaborate before it's too late. A written CDD may occur between some initial prototyping or spike work and the real implementation. However, don't skip the CDD even if the prototype looks good.

Where

1. Write all CDDs in Confluence or link them in Confluence,
2. Add the changedesigndoc label to the Confluence page (using the generic Change Design Doc template in Confluence automatically adds the label), and
3. Place CDDs in a folder under the relevant team or project.

In the new Jira instance, create CDDs primarily for Features and Epics and use the "Design Doc" field to link CDDs to Jira Features (previously known as MMFs) and Epics.

Who

Generally, the stakeholder roles in a Change Design Document are as follows:

- The **Engineering Manager** is accountable for CDD creation. EMs gauge the complexity of work and determine whether a CDD is required.
- The **Engineering Team** proposes solutions and writes the CDD.
- The **Product Manager** adds the business context to the CDD (if necessary) and speaks to the customer impact of the different solutions.
- The **Architect(s)** reviews and approves the CDD but can assist in writing it. They ask for clarification, involve other stakeholders, and escalate as required.

At a minimum, the group Tech Lead and/or Architect should review the CDD. Tag Product Managers for awareness when the change has significant business or functional implications.

How

See full details about the Change Design document process. Try the generic Change Design Doc template available in Confluence. For help writing Change Design Docs or for questions about the process or Confluence template, ask in the #architecture Slack channel.