

Design Reviews

Goals

- Reduce technical debt for our customers
- Continue to iterate on design after Game Plan review
- Generate useful technical artifacts that can be referenced by Microsoft and customers

Measures

Cost of Change

When incorporating design reviews as part of the engineering process, decisions are front-loaded before implementation begins. Making a decision of using Azure Kubernetes Service instead of App Services at the design phase likely only requires updating documentation. However, making this pivot after implementation has started or after a solution is in use is much more costly.

Are these changes occurring before or after implementation? How large of effort are they typically?

Reviewer Participation

How many individuals participate across the designs created? Cumulatively if this is a larger number this would indicate a wider contribution of ideas and

perspectives. A lower number (i.e. same 2 individuals only on every review) might indicate a limited set of perspectives. Is anyone participating from outside the core development team?

Time To Potential Solutions

How long does it typically take to go from requirements to solution options (multiple)?

There is a healthy balancing act between spending too much or too little time evaluating different potential solutions. Too little time puts higher risk of costly changes required after implementation. Too much time delays target value from being delivered; as well as subsequent features in queue. However, the faster the team can *identify the most critical information necessary to make an informed decision*, the faster value can be provided with lower risk of costly changes down the road.

Time to Decisions

How long does it take to make a decision on which solution to implement?

There is also a healthy balancing act in supporting a healthy debate while not hindering the team's delivery. The ideal case is for a team to quickly digest the solution options presented, ask questions, and debate before finally reaching quorum on a particular approach. In cases where no quorum can be reached, the person with the most context on the problem (typically story owner) should make the final decision. Prioritize delivering value and learning. Disagree and commit!

Impact

- Solutions can be quickly be operated into customer's production environment
- Easier for other dev crews to leverage your teams work
- Easier for engineers to ramp up on projects
- Increase team velocity by front-loading changes and decisions when they cost the least
- Increased team engagement and transparency by soliciting wide reviewer participation

Participation

Dev Crew

The dev crew should always participate in all design review sessions

Domain Experts

Domain experts should participate in design review sessions as needed

- ISE Tech Domains
- Customer subject-matter experts (SME)
- Senior Leadership

Facilitation Guidance

Recipes

Please see our [Design Review Recipes](#) for guidance on design process.

Sync Design Reviews via In-Person / Virtual Meetings

Joint meetings with dev crew, subject-matter experts (SMEs) and customer engineers

Async Design Reviews via Pull-Requests

See the [async design review recipe](#) for guidance on facilitating async design reviews. This can be useful for teams that are geographically distributed across different time-zones.

Technical Spike

A technical spike is most often used for evaluating the impact new technology has on the current implementation. Please read more [here](#).

Design Documentation

- Document and update the architecture design in the project design documentation
- Track and document design decisions in a [decision log](#)
- Document decision process in trade studies when multiple solutions exist for the given problem

Early on in engagements, the team must decide where to land artifacts generated from design reviews. Typically, we meet the customer where they are at (for example, using their Confluence instance to land documentation if that is their preferred process). However, similar to storing decision logs, trade studies, etc. in the development repo, there are also large benefits to maintaining design review artifacts in the repo as well. Usually these

artifacts can be further added to root level documentation directory or even at the root of the corresponding project if the repo is monolithic. In adding them to the project repo, these artifacts must similarly be reviewed in Pull Requests (typically preceding but sometimes accompanying implementation) which allows async review/discussion. Furthermore, artifacts can then easily link to other sections of the repo and source code files (via [markdown links](#)).

Last update: August 26, 2024