Systems Diagrams

This method provides systems context for a change – at a high level, what systems are involved and how do they interact?

Description

A systems diagram is a visual model of a system, its components, and their interactions. It describes the necessary inputs and outputs of a system. It is a lower fidelity summary that emphasizes breadth over depth, and doesn’t include human interactions. Developing a systems diagram does not require a high degree of technical skill, just a basic understanding of the components and interactions in a system.

Step by Step

1. Pick a topic: distribution or use of system resources; or system architecture. (Why things are logically arranged such as for re-use)
2. Interview! There’s no way you already know all of the pieces of this puzzle. Identify the experts and pull all of that tacit information out of their heads.
3. Draw boxes for systems and connect them with lines representing their interactions.
4. Slap a brief description on some of the interactions. This could include: what data is being transferred? At what times? Using what technologies? Don’t feel limited or obligated to in what level of detail you provide.
5. Test your model on the Subject Matter Experts from point 1 above! Collect their feedback and revise your model.

Pro tip: Keep it Simple. Don’t overload your document with lots of detail. You’re trying to communicate high level information.

Benefits

- **Understandable system impacts**: systems diagrams show what complex systems interact with to help in determining if a change or development to one area impacts another.
- ** Clearly documented pipelines**
- **Simplified physical architecture**: diagrams can be helpful for documenting the arrangement of physical system elements. (i.e. server names, on-prem/cloud, pipelines are between them, firewalls, etc.)
- **Strange Bedfellows**: systems diagrams are useful for communicating between groups that own distant parts of the same system, and helping them see how they are interdependent.
- **Systems Stakeholders**: diagrams can identify all the internal and external parties involved. (owners of different systems, integrations, reports, services, APIs, etc.)

When to Use

Use it to communicate a lot of information and identify where the information is (not the detail, but the breadth). It is most relevant when there are interactions with many technologies and hand-offs or pipelines between them. This can show your audience impacts from possible changes to the pipelines or hand-offs. Likely uses include communicating from one systems architect to another, or clarifying system service owners to diverse stakeholders. (i.e. where does my responsibility end and yours start)

Example when to use:

- A web form, which interacts with a service bus, that puts records in a table, that is reported on and interacted with through a desktop app. In other words, wherever many technologies are interacting through pipelines. This can be especially useful if there are time dependencies.
- Replacement of a legacy, native table with an external source. Diagrams are useful when something is new or when making a change.

Example when it’s not necessary:

- When depicting simple systems relationships. For example, a single BI visualization or data storage table. (i.e. situations where there aren’t many technologies interacting)

References
• **System Diagram Wiki**: This is a very thorough wiki post [https://deseng.ryerson.ca/dokuwiki/design/system_diagram](https://deseng.ryerson.ca/dokuwiki/design/system_diagram)

• **EIP System Diagram**: This is a UW specific example for the Enterprise Integration Platform: [http://wiki.cac.washington.edu/display/EIP/EIP+System+Diagram](http://wiki.cac.washington.edu/display/EIP/EIP+System+Diagram)

• **Example Diagram**: Here's just a helpful example from StackExchange: