Standards/Guidelines for the Producers of Web Services/APIs to decide the best options for versioning their services
Stephen Brawn and Mark Nuzzo
11-21-2017

Table of Contents

1. Summary Page 2
2. Suggestions for when to create a new version or update an existing version Page 2
3. Guidelines for deprecating an old version Page 3
4. Guidelines for version numbering Page 3
1. **Summary:**

In an effort to standardize how Northwestern IT (NIT) departments determine when and how it is appropriate to roll out new versions of a service or deprecate existing versions, this document offers some guidelines and suggestions. While these are just guidelines, and the decision ultimately lies with the producing department, the goal is that following these standards will help NUIT, and the broader community that we serve, foster some commonalities and enable us to adopt more streamlined processes and service rollouts.

2. **Suggestions for when to create a new version or update an existing version.**

   **Point – to – Point services with only one customer**

   While point-to-point services in our service oriented culture should be rare, there may still be reasons to have them – such as when integrating with vendor systems that require very specific and unique business rules. When creating new services, your approach should always be towards reusability, however, if you do have a need for a dedicated service with only one customer – or have existing services of this type – the versioning conversations is very different than for services with multiple customers – or the potential for multiple customers. For this scenario, it may not be worth the effort to create a new version of the service and inherit the overhead of deprecating an old version. Since there would only be one customer for this service, the most likely scenario is that both the provider and consumer are working hand-in-hand on the changes and updating the current version is the most saleable option.

   **Services with multiple customers – or the potential for multiple customers**

   For services with multiple customers – or the potential for multiple customers – we are recommending that new versions of the service get created by the provider for any substantial change that might require more than minimal development work by the consumer to utilize the service. The basic idea behind this suggestion is that we do not want to hinder the business of our consumers by constantly updating existing service versions that potentially require them to make immediate updates to their source code to properly continue using the service. While the definition of substantial change is left up to the provider, the idea would be any change that would require more than a couple of hours of work by the consumer to continue utilizing the provided service in the same manner as they are accustomed to. By creating a new version of a provided service with schema changes, additional or modified elements, or other changes, we allow the customer to stay on the previous version until they have time to appropriately plan for and integrate the new version into their systems.

4. **Guidelines for deprecating an old version.**

   We are recommending that all old service versions be deprecated and removed from service – no longer provided - within 6 months of a new version being available. While this deprecation time-frame is up to the provider – in concert with opinions from the consumers – we highly
recommend that old service versions get deprecated as soon as possible to ensure that customers are moved to supported service versions in a timely manner.

5. **Guidelines for version numbering.**

We are recommending – for ease of use and standardization – that all versions are numbered using whole numbers only. I.e. Version 1, Version 2, Version 3, etc.