ServiceNow takes product updates pretty seriously, and has released new and improved versions of its software periodically since Summer 2007. The names of these updates once revolved around the release date, but following the Aspen update in 2011, new versions of ServiceNow have been inspired by famous cities and locations.

For ServiceNow veterans looking to take a trip down memory lane, or new users hoping to study up on the platform’s legacy, here is a breakdown of the history of ServiceNow updates, with release notes on each of them:

### Version History

- **Aspen (2011)**
  - Description: The platform’s 10-month development cycle was much simpler for non-developers, without a technical background.

- **Berlin (2014)**
  - Description: The Berlin update was actually in May 2014, not December 2014 as initially stated. The update introduced new project management templates.

- **Calgary (2013)**
  - Description: With the Calgary update, the emphasis was on designing a governance, risk, and compliance feature. Cloud Engine became a standard management attribute.

- **Fiji (2011)**
  - Description: The Fiji update massively increased the functionality of the ServiceNow platform, Geneva was a major update.

- **Geneva (2014)**
  - Description: The Geneva update was in March 2014, not May 2012 as initially stated. The update introduced a new user experience with surely to make the ServiceNow platform more accessible.

- **Helsinki (2012)**
  - Description: With the Helsinki update, the platform was intended to improve the overall user experience with an aim to make the ServiceNow platform more accessible.

- **Jakarta (2015)**
  - Description: The Jakarta version, and its emphasis on machine learning, really unlocked the potential of data analytics inside the platform.

- **Kingston (2017)**
  - Description: With the Kingston version, the emphasis was on designing a governance, risk, and compliance feature. Cloud Engine became a standard management attribute.

- **Dublin (2013)**
  - Description: With the Dublin version, the emphasis was on designing a governance, risk, and compliance feature. Cloud Engine became a standard management attribute.

- **Instanbul (2018)**
  - Description: With the Instanbul version, the emphasis was on designing a governance, risk, and compliance feature. Cloud Engine became a standard management attribute.

- **Jakarta (2015)**
  - Description: The Jakarta version, and its emphasis on machine learning, really unlocked the potential of data analytics inside the platform.

- **Jako (2019)**
  - Description: The Jako version, and its emphasis on machine learning, really unlocked the potential of data analytics inside the platform.

- **Kungston (2017)**
  - Description: With the Kingston version, the emphasis was on designing a governance, risk, and compliance feature. Cloud Engine became a standard management attribute.

- **Dublin (2013)**
  - Description: With the Dublin version, the emphasis was on designing a governance, risk, and compliance feature. Cloud Engine became a standard management attribute.

- **Instanbul (2018)**
  - Description: With the Instanbul version, the emphasis was on designing a governance, risk, and compliance feature. Cloud Engine became a standard management attribute.

- **Jakarta (2015)**
  - Description: The Jakarta version, and its emphasis on machine learning, really unlocked the potential of data analytics inside the platform.

- **Jako (2019)**
  - Description: The Jako version, and its emphasis on machine learning, really unlocked the potential of data analytics inside the platform.

- **Kungston (2017)**
  - Description: With the Kingston version, the emphasis was on designing a governance, risk, and compliance feature. Cloud Engine became a standard management attribute.

- **Dublin (2013)**
  - Description: With the Dublin version, the emphasis was on designing a governance, risk, and compliance feature. Cloud Engine became a standard management attribute.

- **Instanbul (2018)**
  - Description: With the Instanbul version, the emphasis was on designing a governance, risk, and compliance feature. Cloud Engine became a standard management attribute.

- **Jakarta (2015)**
  - Description: The Jakarta version, and its emphasis on machine learning, really unlocked the potential of data analytics inside the platform.

- **Jako (2019)**
  - Description: The Jako version, and its emphasis on machine learning, really unlocked the potential of data analytics inside the platform.

- **Kungston (2017)**
  - Description: With the Kingston version, the emphasis was on designing a governance, risk, and compliance feature. Cloud Engine became a standard management attribute.

- **Dublin (2013)**
  - Description: With the Dublin version, the emphasis was on designing a governance, risk, and compliance feature. Cloud Engine became a standard management attribute.

- **Instanbul (2018)**
  - Description: With the Instanbul version, the emphasis was on designing a governance, risk, and compliance feature. Cloud Engine became a standard management attribute.

- **Jakarta (2015)**
  - Description: The Jakarta version, and its emphasis on machine learning, really unlocked the potential of data analytics inside the platform.

- **Jako (2019)**
  - Description: The Jako version, and its emphasis on machine learning, really unlocked the potential of data analytics inside the platform.

- **Kungston (2017)**
  - Description: With the Kingston version, the emphasis was on designing a governance, risk, and compliance feature. Cloud Engine became a standard management attribute.

- **Dublin (2013)**
  - Description: With the Dublin version, the emphasis was on designing a governance, risk, and compliance feature. Cloud Engine became a standard management attribute.

- **Instanbul (2018)**
  - Description: With the Instanbul version, the emphasis was on designing a governance, risk, and compliance feature. Cloud Engine became a standard management attribute.

- **Jakarta (2015)**
  - Description: The Jakarta version, and its emphasis on machine learning, really unlocked the potential of data analytics inside the platform.

- **Jako (2019)**
  - Description: The Jako version, and its emphasis on machine learning, really unlocked the potential of data analytics inside the platform.

- **Kungston (2017)**
  - Description: With the Kingston version, the emphasis was on designing a governance, risk, and compliance feature. Cloud Engine became a standard management attribute.

- **Dublin (2013)**
  - Description: With the Dublin version, the emphasis was on designing a governance, risk, and compliance feature. Cloud Engine became a standard management attribute.

- **Instanbul (2018)**
  - Description: With the Instanbul version, the emphasis was on designing a governance, risk, and compliance feature. Cloud Engine became a standard management attribute.

- **Jakarta (2015)**
  - Description: The Jakarta version, and its emphasis on machine learning, really unlocked the potential of data analytics inside the platform.

- **Jako (2019)**
  - Description: The Jako version, and its emphasis on machine learning, really unlocked the potential of data analytics inside the platform.

- **Kungston (2017)**
  - Description: With the Kingston version, the emphasis was on designing a governance, risk, and compliance feature. Cloud Engine became a standard management attribute.

- **Dublin (2013)**
  - Description: With the Dublin version, the emphasis was on designing a governance, risk, and compliance feature. Cloud Engine became a standard management attribute.

- **Instanbul (2018)**
  - Description: With the Instanbul version, the emphasis was on designing a governance, risk, and compliance feature. Cloud Engine became a standard management attribute.