

- Welche der gängigen Middleware-Technologien ist für eine bestimmte Aufgabe besser geeignet?**
- Welche Kriterien sind bei der Auswahl einer Architektur zu beachten?**
- Welche Middleware setze ich ein, um eine konkrete Aufgabe schnell, zuverlässig und fehlerfrei zu lösen?**
- Soll ich die Middleware verwenden, die ich bereits kenne?**
- Wie groß ist der Aufwand, wenn ich mich für eine für mich neue Middleware entscheide?**

Diese Fragen beschäftigen Software-Architekt/inn/en oder -Entwickler/innen zu Beginn jedes Projektes. Wird die falsche Entscheidung getroffen, vergrößert sich der Aufwand der Entwicklungsarbeit stark.

Um die Entscheidungsfindung zu erleichtern, wurde ein Technologievergleich durchgeführt, für den ein konkretes Programmierbeispiel mit vier Technologien aus der „.NET-Welt“ gelöst wurde:

- **XVSM**
- **.NET Remoting**
- **WCF**
- **ASP.NET**

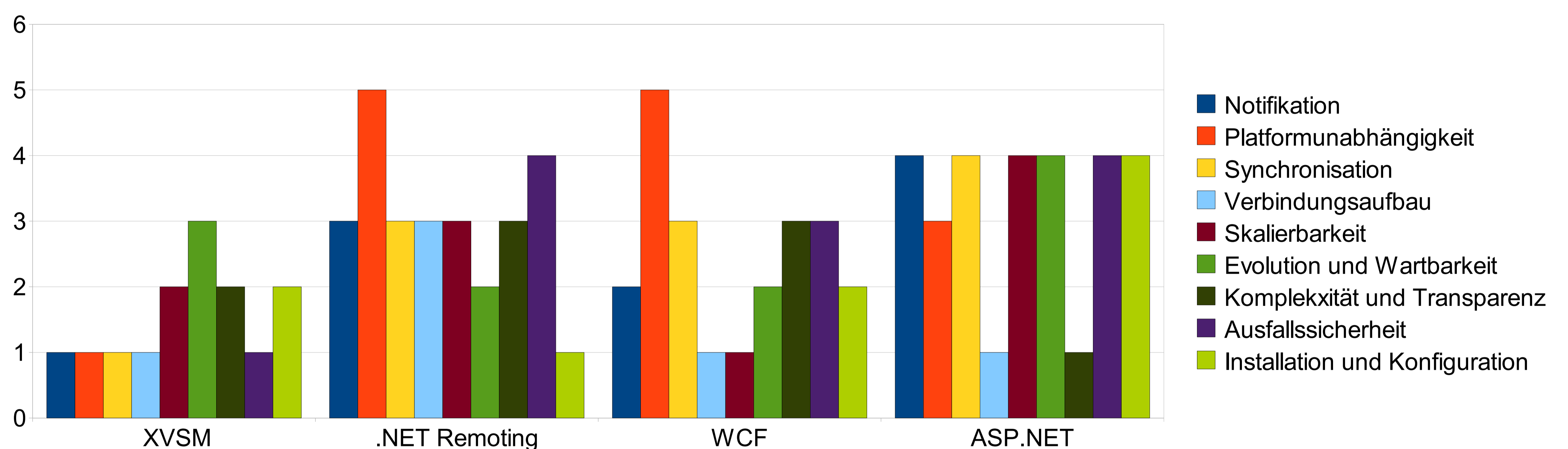


Abbildung 1: Technologievergleich

- **Notifikationen:** Entwicklung von EDA Systemen kann genutzt werden
- **Synchronisation:** gemeinsamer Zugriff von Prozessen auf Daten-Objekte
- **Plattformunabhängigkeit:** für verschiedene Plattformen und in verschiedenen Programmiersprachen
- **Verbindungsaufbau:** Aufwand beim Aufbau einer neuen Verbindung
- **Skalierbarkeit:** Ressourcenbedarf bei wachsenden Eingabemengen
- **Evolution und Wartbarkeit:** wie sehr entspricht das System den Agilen Prinzipien.
- **Komplexität und Transparenz der Technologie:** Verständlichkeit und Abstraktionsniveau
- **Ausfallsicherheit:** Datenverlust bei unerwarteten Störungen
- **Installation und Konfiguration der Plattform:** Installation und Standard-Konfigurationen
- **LOC:** Lines of Code: Anzahl der funktionalen Codezeilen

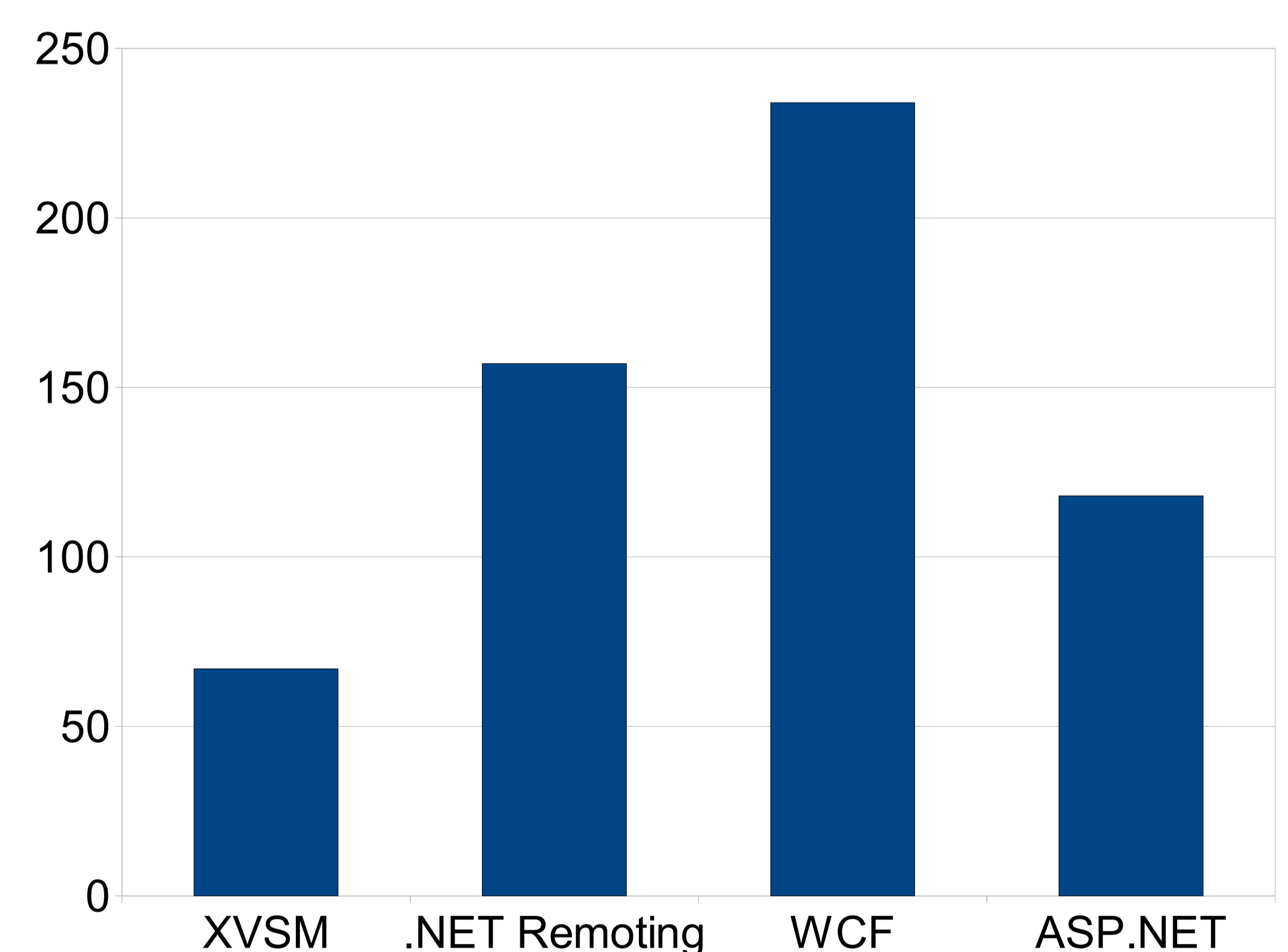


Abbildung 2: Lines-of-Code

Diese neun Kriterien wurden zur Bestimmung von Komplexität und Aufwand beim Programmieren festgelegt und mittels einer Skala von 1 („geringer Aufwand, geringe Komplexität“) bis 5 („großer Aufwand, hohe Komplexität“) bewertet (siehe Abb. 1, LOC siehe Abb.2).

Die Ergebnisse zeigen, dass die XVSM-Technologie (eXtensible Virtual Shared Memory, www.xvsm.org) für die Entwicklung von Verteilten Systemen mit eventbasierten Architekturen sehr gut geeignet ist