



Mówi Ipek:

Linie są także obywatelami pierwszej klasy!

Ipek Ozkaya, seniorka w zespole technicznym w Software Engineering Institute w Carnegie Mellon University

Moja praca polega na pomaganiu organizacjom i zespołom w poprawianiu jakości ich systemów z punktu widzenia sprawności ich architektury. Nieuniknioną i oczywistą prośbą w tych działaniach jest „Pokaż mi swoją architekturę”. Przez lata, opierając się na otrzymanych odpowiedziach, opracowałam własny katalog nieporozumień dotyczących architektury i jej tworzenia.

Wydruk wszystkich diagramów sekwencji dla wszystkich scenariuszy użycia, które aż do tej pory pojawiły się w twoich myślach, nie jest twoją architekturą! Chociaż zbiór wszystkich przypadków użycia i ich sposobów zachowania, czyli tego, co umieszczasz w diagramach sekwencji, jest użyteczny i ważny dla twojego systemu, to nie zapewnia on odpowiedniego poziomu abstrakcji dla wnioskowania o wariantach zachowania systemu.

Przegląd kodu nie może zastąpić przeglądu architektury! Najważniejszy cel każdej pracy związanej z architekturą to zaprojektowanie i wdrożenie systemu spełniającego cele biznesowe i cele interesariuszy. Działający kod jest nieuchronną rzeczywistością. Jednak potrzeby architektoniczne przecinają się z wieloma elementami w zaimplementowanym systemie. Efektywny przegląd architektury wiąże się z tymi wymaganiami istotnymi dla architektury oraz wszystkimi elementami, których dotyczą. Tradycyjne praktyki sprawdzania kodu nie obejmują tej całościowej perspektywy.

Pudełka nie są jedynymi elementami architektonicznymi! W przypadkach gdy zespół ma artefakt, często nazywany dokumentacją architektury oprogramowania, i niestety nazywany również SAD, to zawiera on opisy rysunków *ad hoc* składających się z linii i pudełek. To świetny początek, ale długie dyskusje koncentrują się na pudełkach, podczas gdy linie bywają całkowicie zapomniane. To niefortunne, ponieważ w wielu przypadkach linie opisują najważniejsze aspekty decyzji architektonicznych.

Pośród tych trzech nieporozumień najważniejsze to docenienie znaczenia linii na diagramach architektury oprogramowania. Pomyśl o tym. Jeśli chcesz zwiększyć wydajność, skup się na częstości i ilości połączeń między elementami. Jeśli chcesz poprawić łatwość modyfikowania, ogranicz interakcje między elementami. Jeśli chcesz zoptymalizować bezpieczeństwo, chroń relacje między elementami. Wszystko to reprezentują linie!