Guy Steele discusses the key components that go into developing a language, and the approach that allows for a complete yet concise one. The style of the talk was a step by step approach of growing a new language by starting with bare necessities and branching on these where more requirements surface. It worked well because the idea he was conveying was to start small when developing a new language and let the users/developers add to it. For example, he begins his new language with single syllable words, and defines multiple syllable words based off each other. A language shouldn’t be confined to being large or small, but rather it should strive to be the most universal language while keeping it as small as possible. So, the smaller the better, but do not sacrifice necessities for simplicity. A metaphor that stuck out to me was implementing a Bazaar-like language with a lead architect. A bazaar that does not have a lead architect can be very chaotic, but when each stall works through the same “lead”, different parts/stalls can be integrated in while avoiding the issue of compatibility. His high point began with a quote from Christopher Alexander about the use of a master plan. The existence of a master plan essentially alienates the users from having any presence or impact on the shape of the language or, in the quote, their community. They simply will feel like a cog in a wheel rather than having a sense of identification or purpose. Guy Steele responds to Christopher Alexander’s quote by saying, “But instead of designing a thing, you need to design a way of doing.” He means develop a process of going about additions and integration, rather than laying down the law on how the language will look. Taken that this video is from 1998, it is actually very interesting to see how relevant it is today. For example, he discussed a means of being able to create your own definitions and functions through “lambda”, and Java implemented it in Java 8, while current versions of multiple dominant languages allow the use of lambda, or defining functions as you go. Plus, many languages use the bazaar like process through pods (with iOS) and packages that are developed by normal users. I guess this talk really helped me understand the approach of integrating a language by starting small and building up when needed. Also, a language is more collaborative and molds to how the users need it. I definitely think this video was worth the watch and will help change the perspective of future students.