Go *does* have address the issues that make large-scale software development difficult. These issues include:

These issues include:

* slow builds
* uncontrolled dependencies
* each programmer using a different subset of the language
* poor program understanding (code hard to read, poorly documented, and so on)
* duplication of effort
* cost of updates
* version skew
* difficulty of writing automatic tools
* cross-language builds

与C/C++依赖性

通过使用#ifndef保护器确实可以保护程序但是对软件的影响是C程序中#include子句的逐渐积累。添加它们不会破坏程序，很难知道何时不再需要它们。删除一个#include并再次编译该程序甚至不足以测试它，因为另一个#include本身可能包含一个#include，无论如何它都会将其拉出计划9方法的最重要结果是编译速度大大提高：与使用带有#ifndef保护器的库编译程序时相比，编译所需的I / O数量可以大大减少。

**语义**

Go语句的语义通常类似于C。 它是一种带有指针等的编译的，静态类型的过程语言。 通过设计，熟悉C系列语言的程序员应该熟悉它。

**并发**

并发对于现代计算环境很重要，它的多核计算机运行具有多个客户端的Web服务器，这可以称为典型的Google程序。 C ++或Java不能很好地支持这种软件，因为C ++或Java在语言级别上缺乏足够的并发支持。

Go体现了具有一流渠道的CSP的变体。 选择CSP的部分原因是出于熟悉程度（我们中的一个人曾在CSP的思想基础上开发过以前的语言），还因为CSP具有易于添加到过程编程模型的特性，而无需对该模型进行深刻的更改。 也就是说，在给定类C语言的情况下，CSP可以以一种几乎正交的方式添加到该语言中，从而提供了额外的表达能力而又不限制该语言的其他用途。 简而言之，语言的其余部分可以保持“普通”。

因此，该方法是独立执行常规程序代码的功能的组合。