



# **Введение в системное программирование**



## Системное программирование

*Что это за понятие?*



## Системное программирование

*Что это за понятие?*

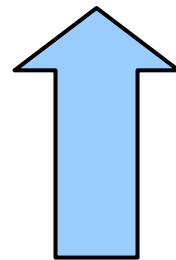
**Процесс конструирования  
программных систем**



## Системное программирование

*Что это за понятие?*

Процесс конструирования  
**программных систем**



*А это что такое?*



## ТО К ЧЕМУ МЫ ПРИВЫКЛИ

Программа

- ✓ Разработана индивидуально
- ✓ Решает частную задачу



## ТО К ЧЕМУ МЫ ПРИВЫКЛИ

Программа

- ✓ Разработана индивидуально
- ✓ Решает частную задачу
- ✓ Пригодна для запуска автором и его ближайшим окружением
- ✓ Для работы требует той же системы, в которой была разработана



## ТО К ЧЕМУ МЫ ПРИВЫКЛИ

Программа

- ✓ Разработана индивидуально
- ✓ Решает частную задачу
- ✓ Пригодна для запуска автором и его ближайшим окружением
- ✓ Для работы требует той же системы, в которой была разработана
- ✓ Простая не значит плохая! В большинстве случаев вполне устраивает



Программа



**Дополнительные  
требования:**





Программа



**Дополнительные  
требования:**

- ✓ Разработка в **обобщённом стиле**: без привязки к системе, форматам, значениям параметров



Программа



## Дополнительные требования:

- ✓ Разработка в **обобщённом стиле**: без привязки к системе, форматам, значениям параметров
- ✓ Требование **надёжности**: критический сбой есть проблема автора, а не пользователя



Программа



## Дополнительные требования:

- ✓ Разработка в **обобщённом стиле**: без привязки к системе, форматам, значениям параметров
- ✓ Требование **надёжности**: критический сбой есть проблема автора, а не пользователя
- ✓ Наличие полной и удобной для использования **документации**



Программа

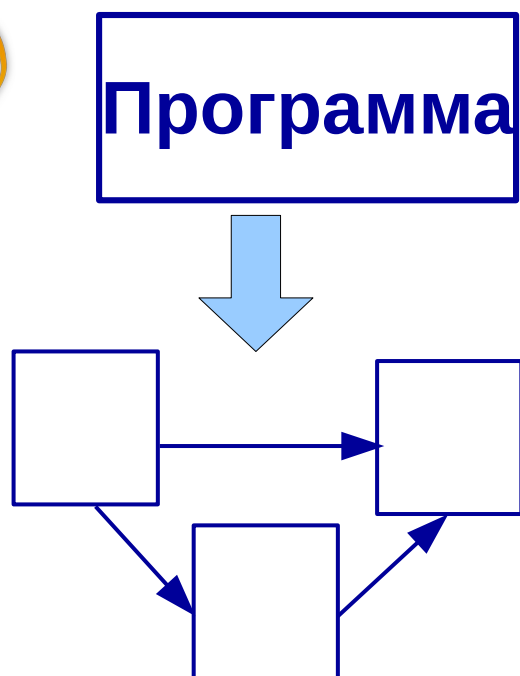


## Дополнительные требования:

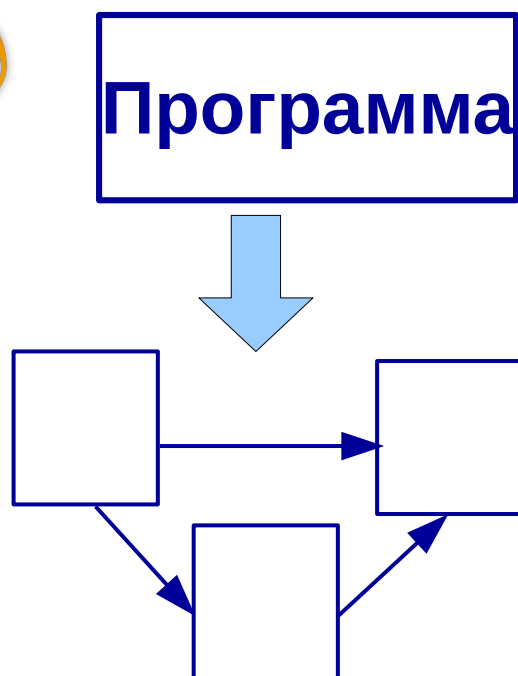
- ✓ Разработка в **обобщённом стиле**: без привязки к системе, форматам, значениям параметров
- ✓ Требование **надёжности**: критический сбой есть проблема автора, а не пользователя
- ✓ Наличие полной и удобной для использования **документации**
- ✓ Обеспечение **сопровождения** — устранение ошибок, расширение функциональности.



# Программный комплекс

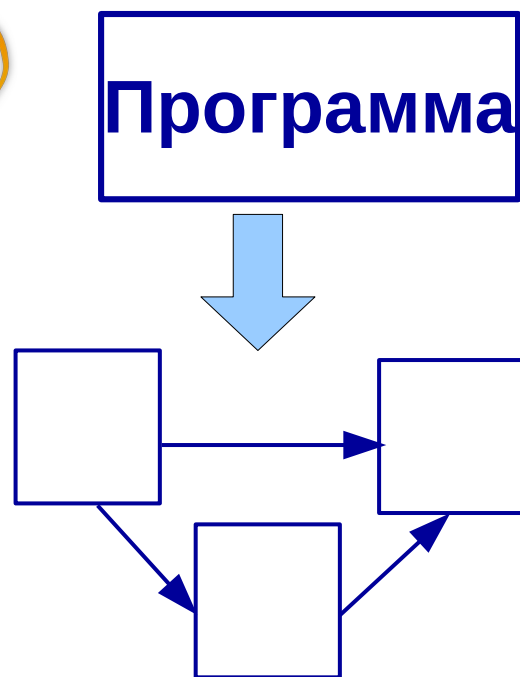


**Дополнительные требования:**



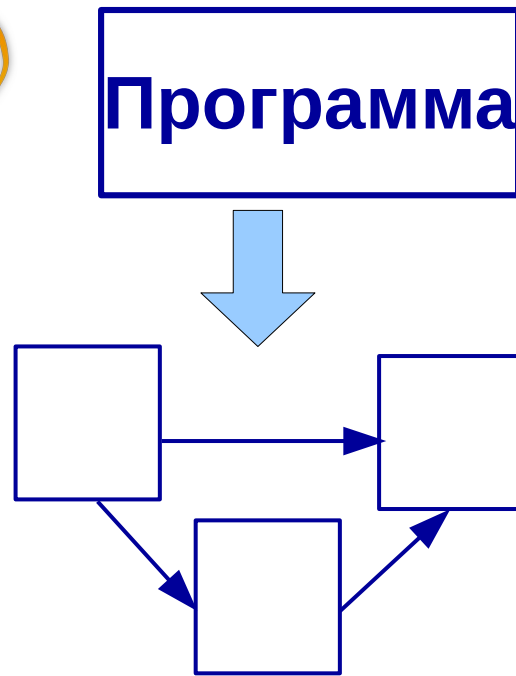
**Дополнительные требования:**

- ✓ **Согласование компонент по функциям**



**Дополнительные требования:**

- ✓ Согласование компонент по **функциям**
- ✓ Согласование по **форматам данных**



## Дополнительные требования:

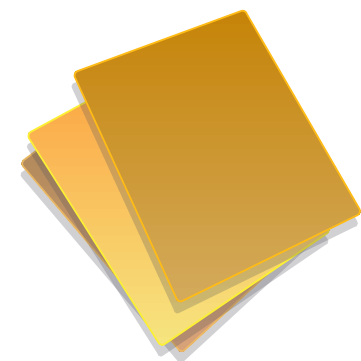
- ✓ Согласование компонент по **функциям**
- ✓ Согласование по **форматам данных**
- ✓ К функциональному тестированию компонент добавляется **системное тестирование** комплекса.





# Системный программный продукт

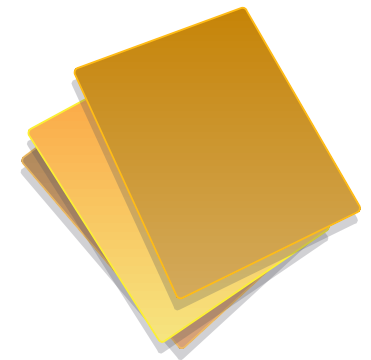
- ✓ Комплекс взаимодействующих программ, надлежащим образом написанный, протестированный и задокументированный с обеспечением сопровождения в процессе его применения пользователями.





# Системный программный продукт

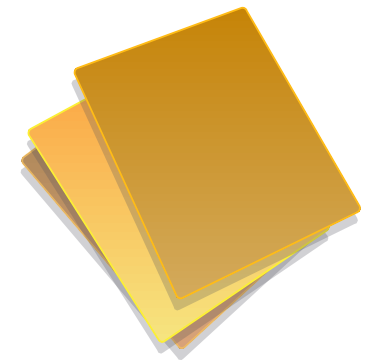
- ✓ Комплекс взаимодействующих программ, надлежащим образом написанный, протестированный и задокументированный с обеспечением сопровождения в процессе его применения пользователями.
- ✓ **Системное программирование** – процесс разработки сложных программных систем в целом или их отдельных компонент с использованием специальных средств, инструментов, парадигм программирования и моделей разработки.

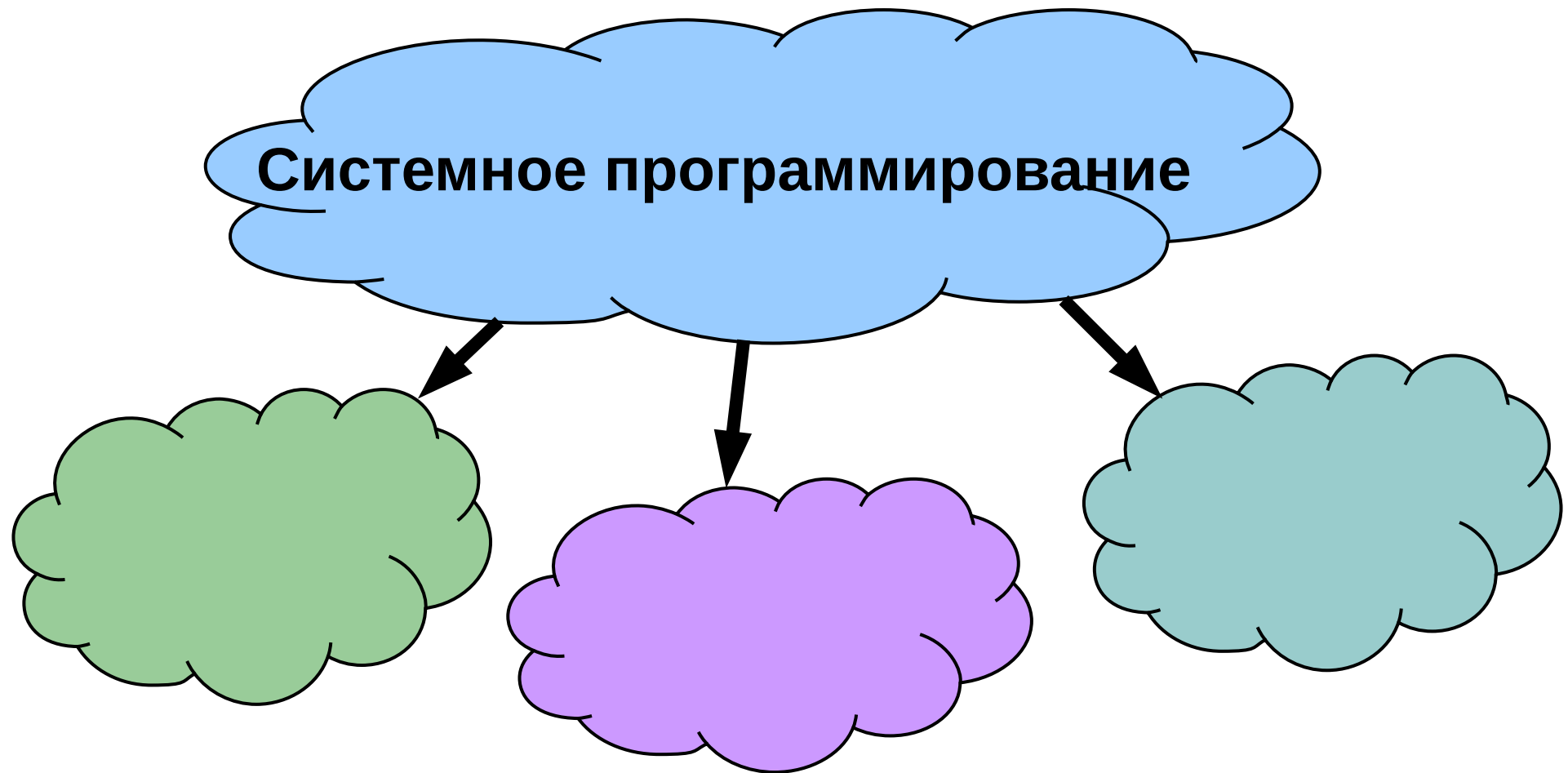


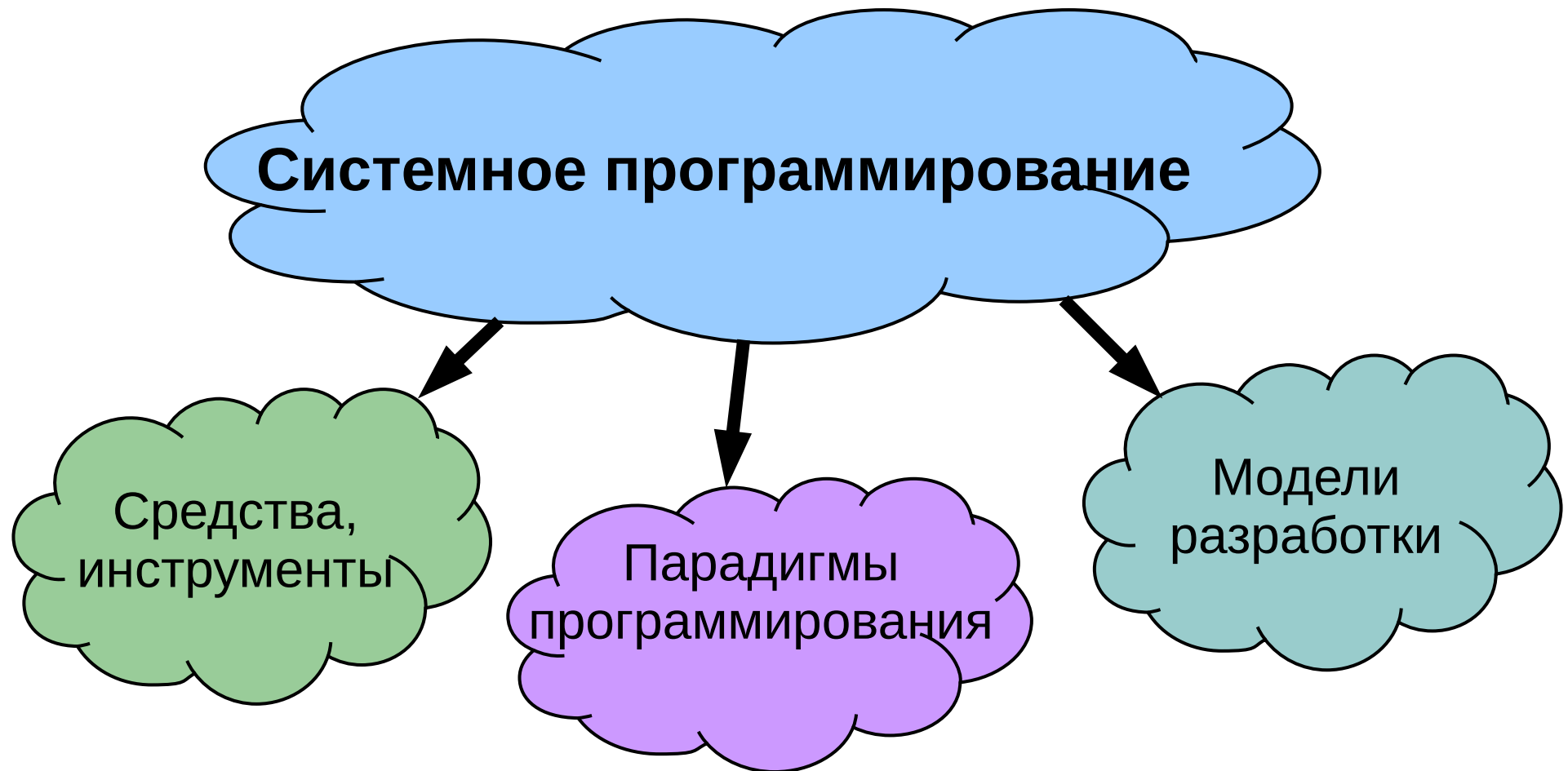


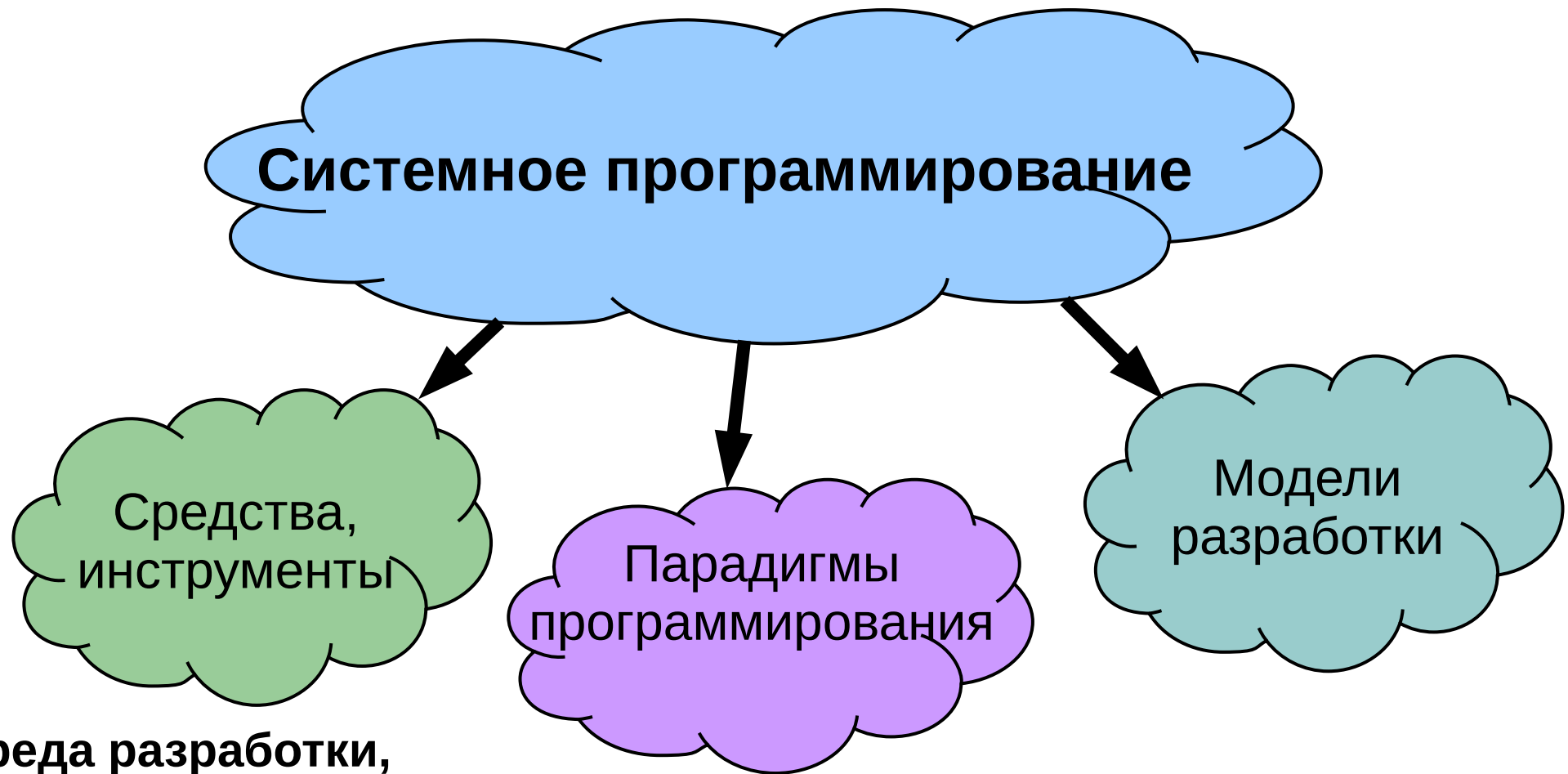
# Системный программный продукт

- ✓ Комплекс взаимодействующих программ, надлежащим образом написанный, протестированный и задокументированный с обеспечением сопровождения в процессе его применения пользователями.
- ✓ **Системное программирование** – процесс разработки сложных программных систем в целом или их отдельных компонент с использованием специальных средств, инструментов, парадигм программирования и моделей разработки.









Среда разработки,  
язык программирования,  
библиотеки



## Системное программирование

Средства,  
инструменты

Среда разработки,  
язык программирования,  
библиотеки

Парадигмы  
программирования

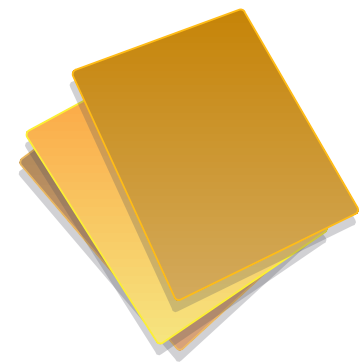
Модели  
разработки

?

?



- ✓ **Парадигма** — совокупность идей и понятий, определяющая стиль программирования.



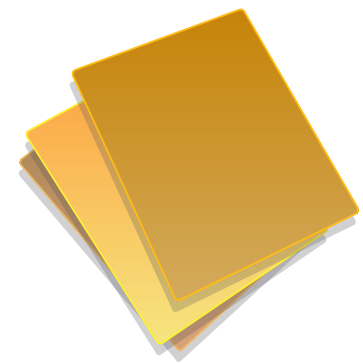




# Парадигма программирования



- ✓ **Парадигма** — совокупность идей и понятий, определяющая стиль программирования.
- ✓ Определяет в каких терминах описывается логика программы.
- ✓ Многие языки программирования поддерживают несколько парадигм.



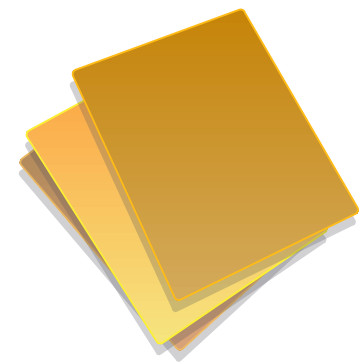


# Парадигма программирования



- ✓ **Парадигма** — совокупность идей и понятий, определяющая стиль программирования.
- ✓ Определяет в каких терминах описывается логика программы.
- ✓ Многие языки программирования поддерживают несколько парадигм.

- ✓ **Императивное**
- ✓ **Объектно-ориентированное**
- ✓ **Обобщенное**
- ✓ **Защитное**
- ✓ **Функциональное**
- ✓ **Логическое**

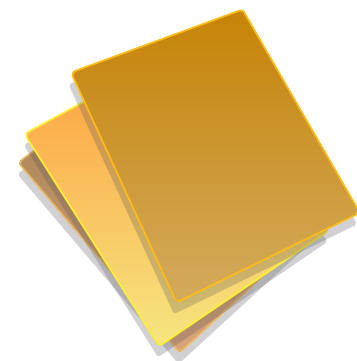




# Модель разработки



- ✓ **Модель разработки** — определяет каким образом будут выполняться действия по разработке программной системы, посредством описания «последовательности» этих действий.

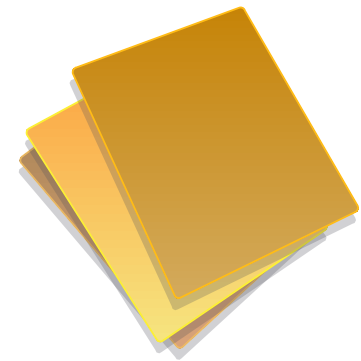




# Модель разработки



- ✓ **Модель разработки** — определяет каким образом будут выполняться действия по разработке программной системы, посредством описания «последовательности» этих действий.
- ✓ Определяет логику прохождения стадий разработки.
- ✓ Не привязывается к конкретным языкам программирования.



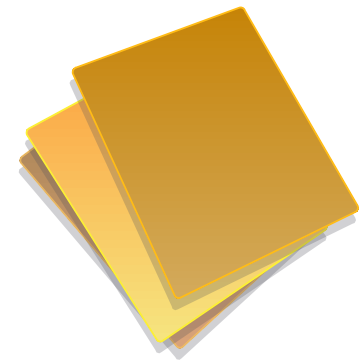


# Модель разработки



- ✓ **Модель разработки** — определяет каким образом будут выполняться действия по разработке программной системы, посредством описания «последовательности» этих действий.
- ✓ Определяет логику прохождения стадий разработки.
- ✓ Не привязывается к конкретным языкам программирования.

- ✓ **Сверху-вниз**
- ✓ **Каскадная**
- ✓ **Спиральная**
- ✓ **Инкрементная**
- ✓ **RAD, визуальное**
- ✓ **XP (экстремальная)**





**Главный Технический Императив Разработки ПО:**

**Управление  
Сложностью**



## НАМ НЕ ПОВЕЗЛО



✓ Простые задачи решены до нас



## НАМ НЕ ПОВЕЗЛО



- ✓ Простые задачи решены до нас
- ✓ Объём программного кода сложных задач — 100 тыс. и миллионы строк исходного кода на языке высокого уровня





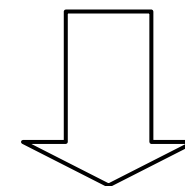
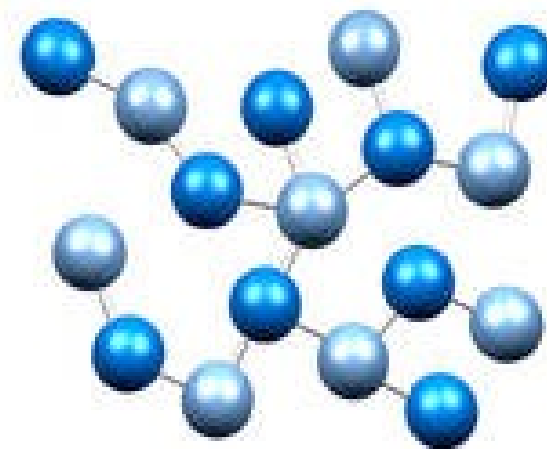
## НАМ НЕ ПОВЕЗЛО



- ✓ Простые задачи решены до нас
- ✓ Объём программного кода сложных задач — 100 тыс. и миллионы строк исходного кода на языке высокого уровня
- ✓ Один разработчик не в состоянии охватить все аспекты подобной программы.

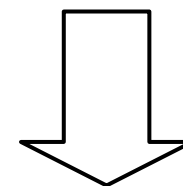
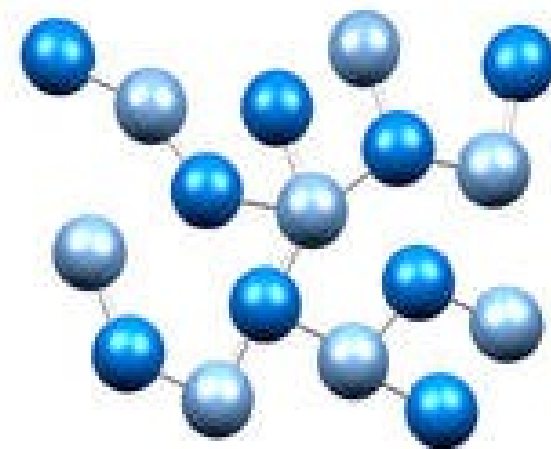


- ✓ Со сложностью можно попытаться справиться, но избавиться от неё нельзя.



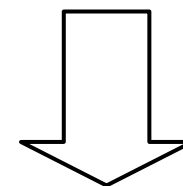
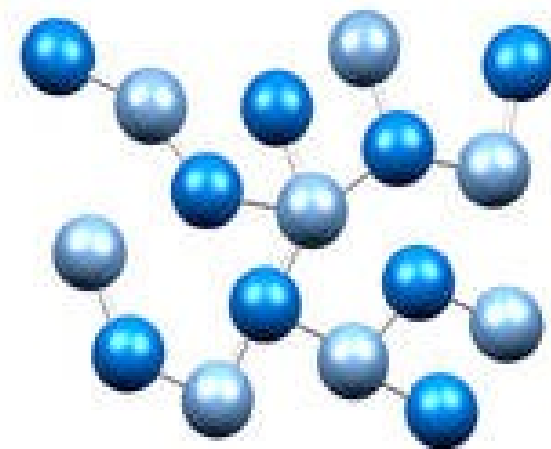


- ✓ Со сложностью можно попытаться справиться, но избавиться от неё нельзя.
- ✓ Чем сложнее система, тем легче её развалить



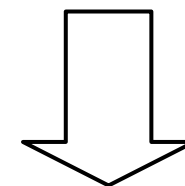
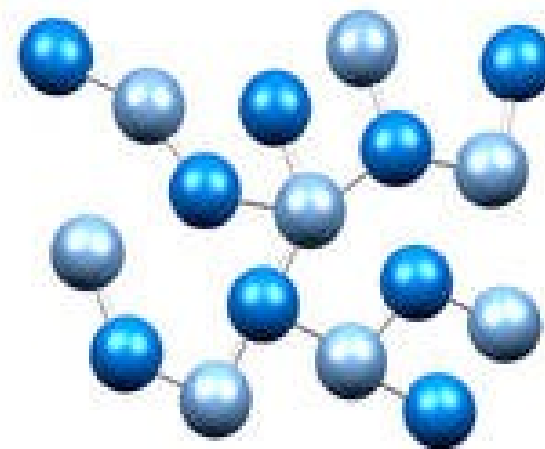


- ✓ Со сложностью можно попытаться справиться, но избавиться от неё нельзя.
- ✓ Чем сложнее система, тем легче её развалить
- ✓ Объем программных систем – **проявление сложности**, но не причина.





- ✓ Со сложностью можно попытаться справиться, но избавиться от неё нельзя.
- ✓ Чем сложнее система, тем легче её развалить
- ✓ Объем программных систем – **проявление сложности**, но не причина.



## А ПОЧЕМУ ОНО ТАКОЕ?

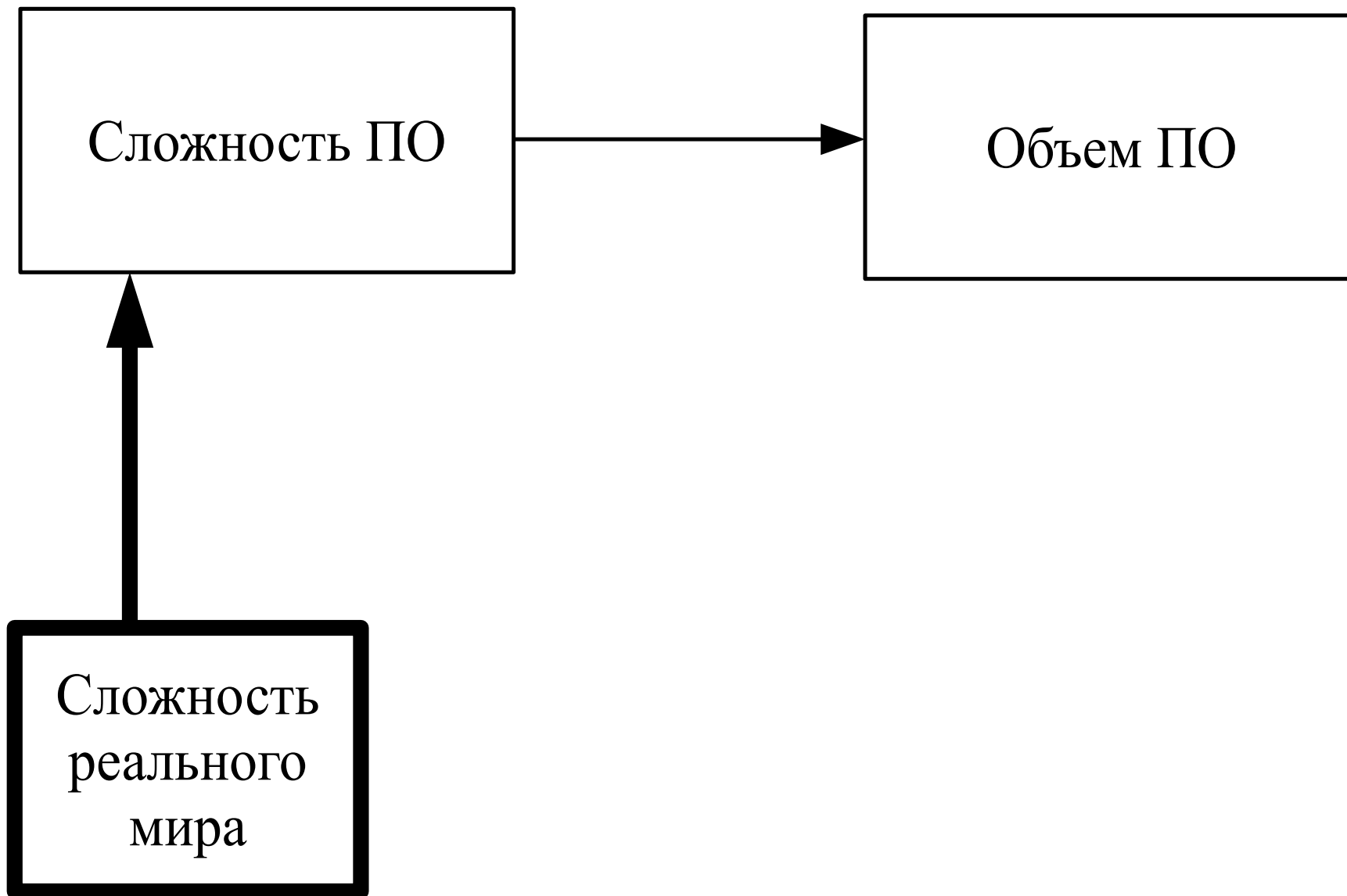


# Причины сложности



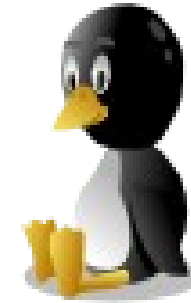
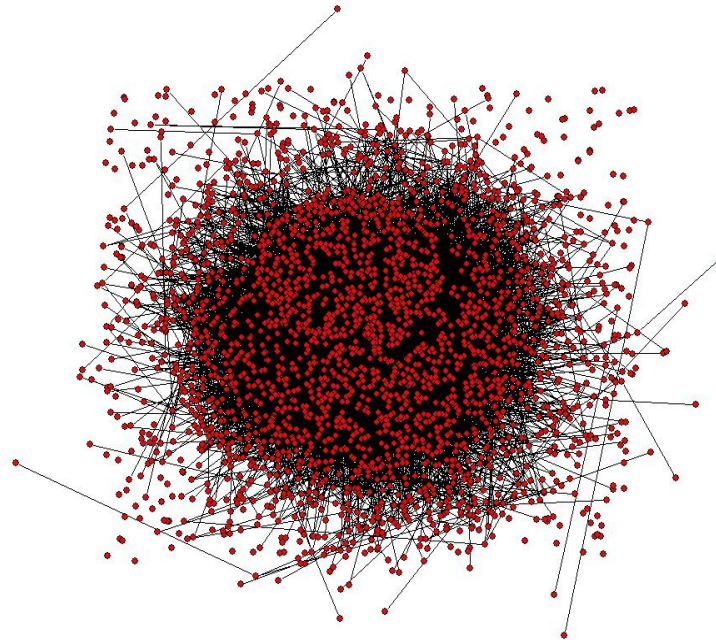
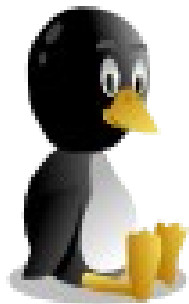


# Причины сложности





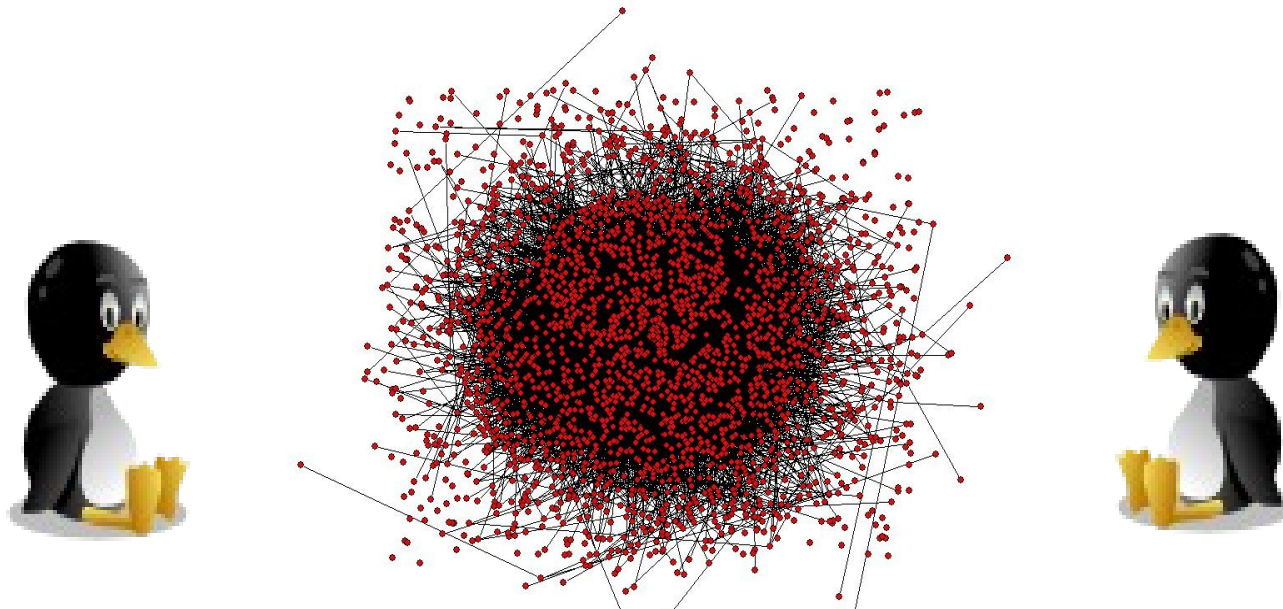
# Сложность реального мира







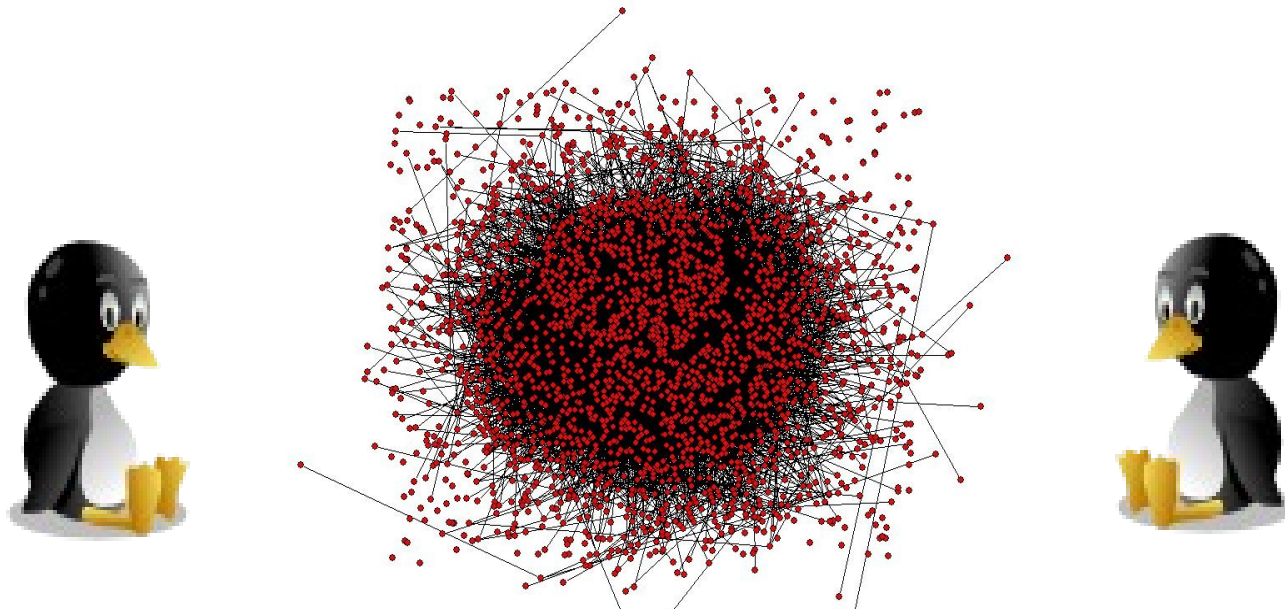
# Сложность реального мира



✓ Не знаем чего хотим (**неполнота требований**)



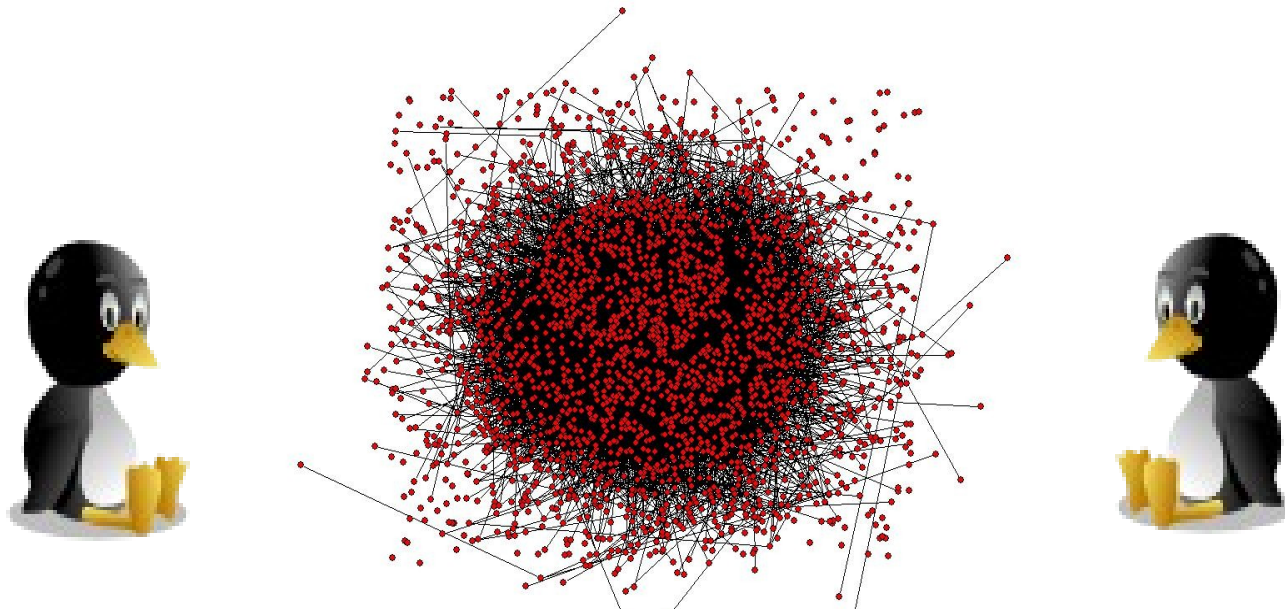
# Сложность реального мира



- ✓ Не знаем чего хотим (**неполнота требований**)
- ✓ **Нечёткость** функциональных требований



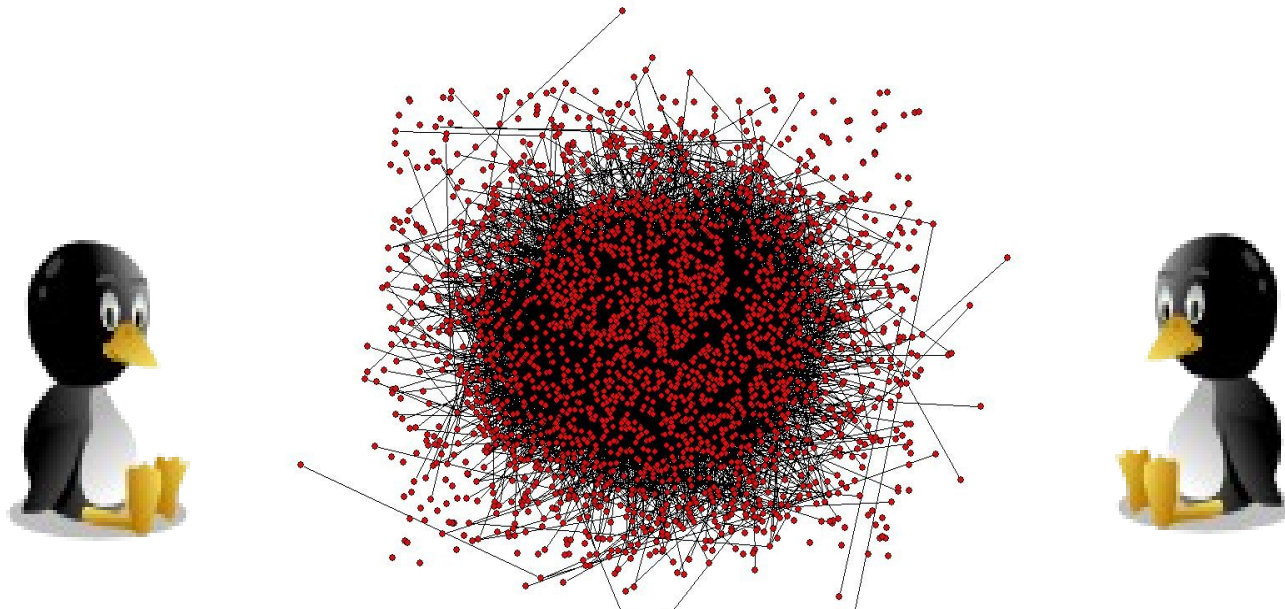
# Сложность реального мира



- ✓ Не знаем чего хотим (**неполнота требований**)
- ✓ **Нечёткость** функциональных требований
- ✓ **Субъективность** эргономических требований



# Сложность реального мира

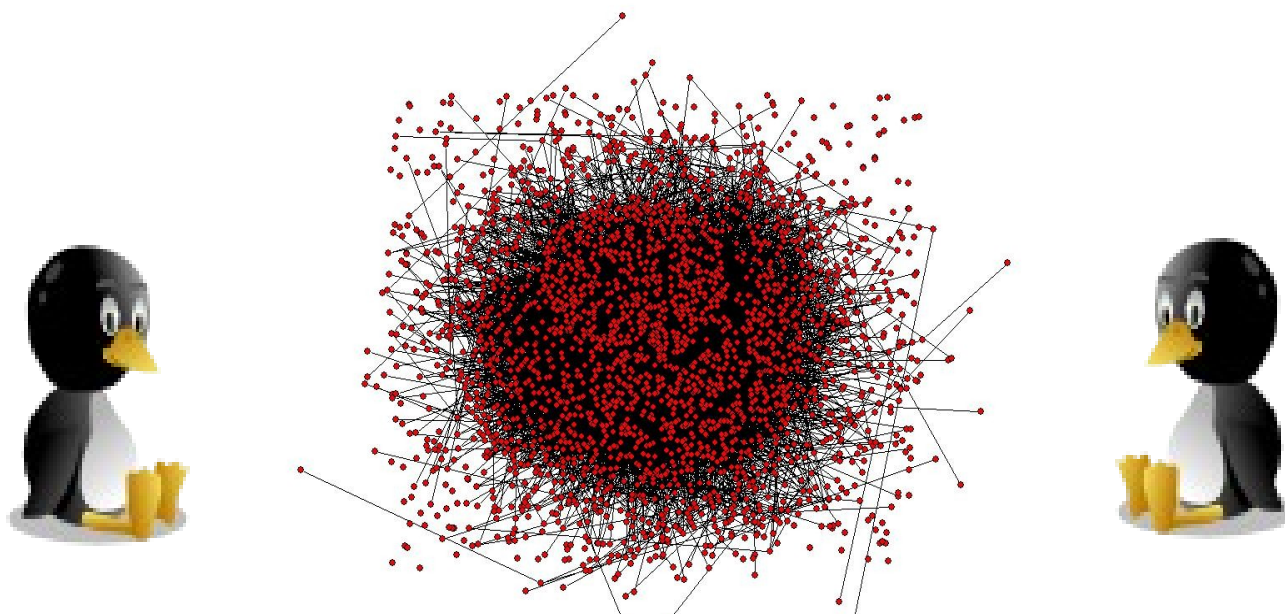


- ✓ Не знаем чего хотим (**неполнота требований**)
- ✓ **Нечёткость** функциональных требований
- ✓ **Субъективность** эргономических требований
- ✓ **Проблема взаимопонимания** между заказчиком и разработчиком

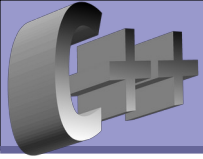




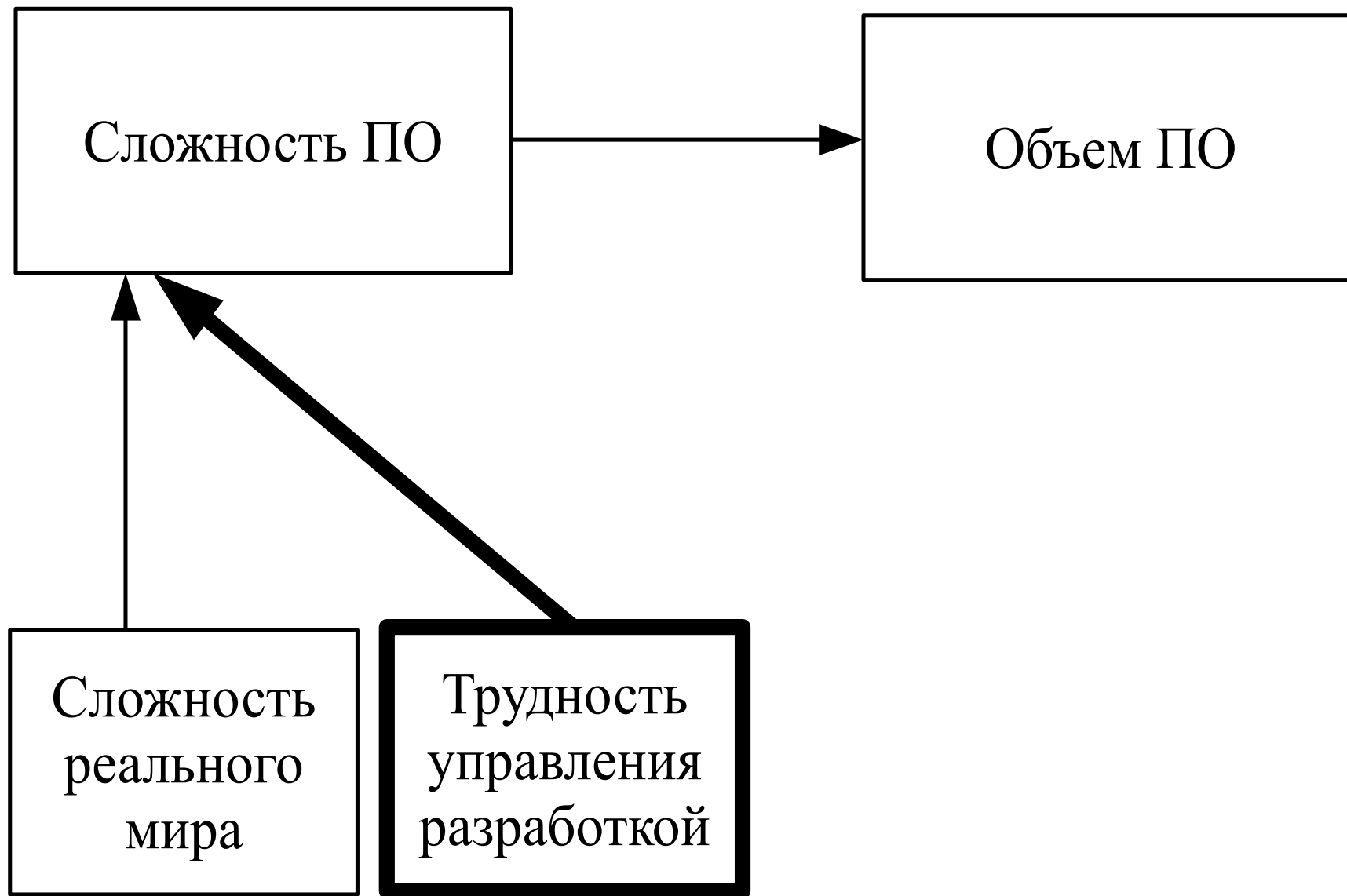
# Сложность реального мира



- ✓ Не знаем чего хотим (**неполнота требований**)
- ✓ **Нечёткость** функциональных требований
- ✓ **Субъективность** эргономических требований
- ✓ **Проблема взаимопонимания** между заказчиком и разработчиком
- ✓ **Смена правил** по ходу игры (гибкость требований)

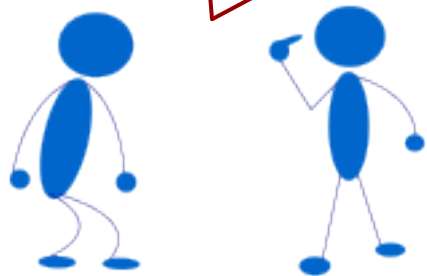


# Причины сложности



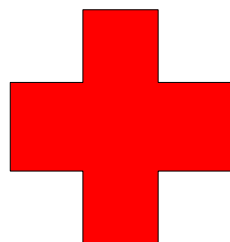


Караул!  
мы не успеваем





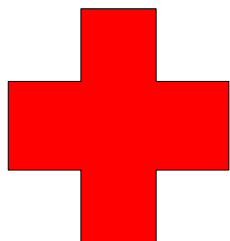
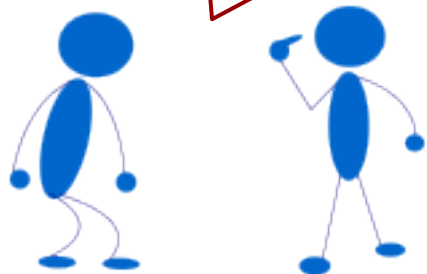
Караул!  
мы не успеваем







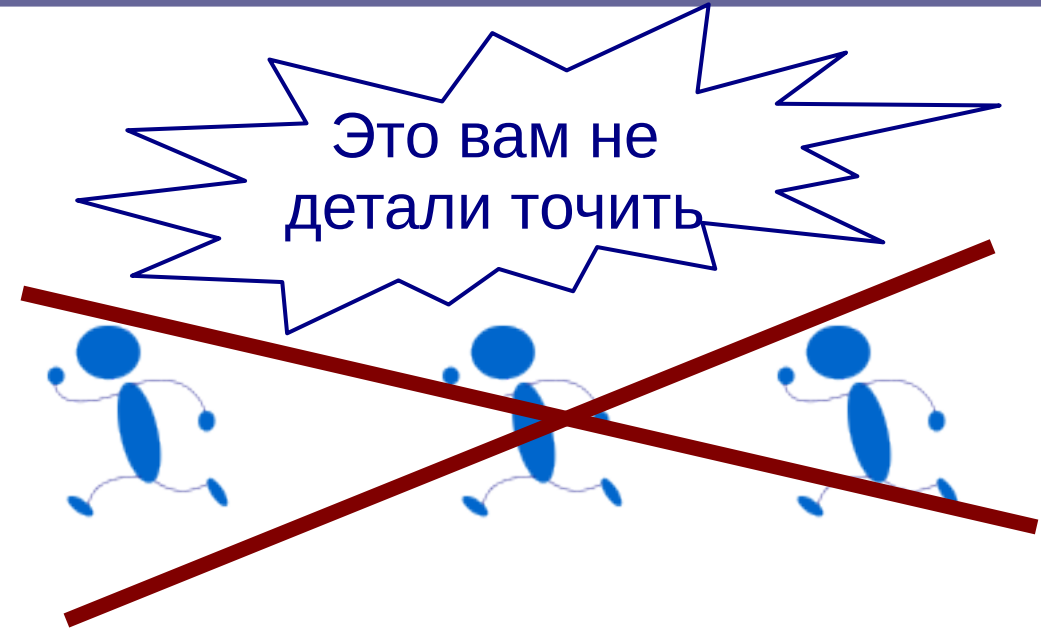
Караул!  
мы не успеваем



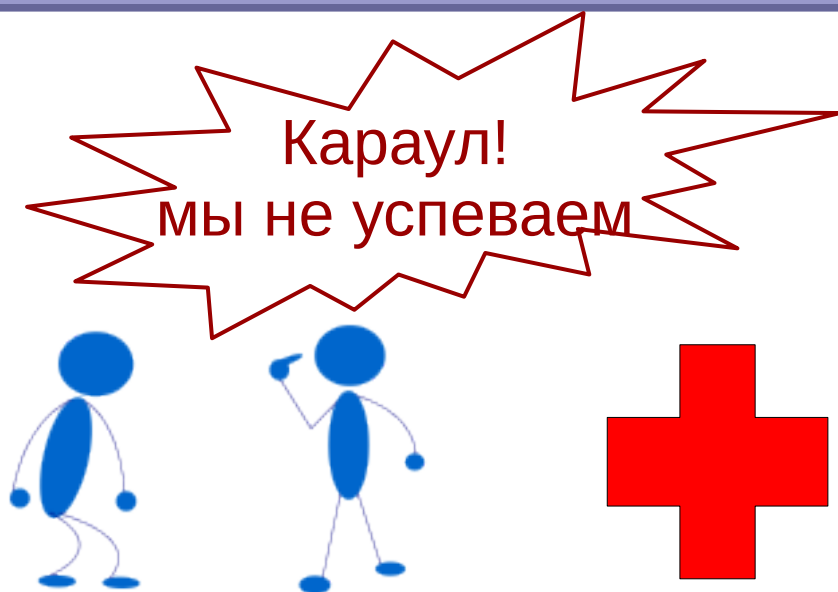
Это вам не  
детали точить



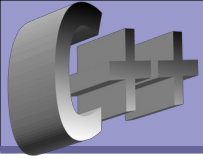
✓ Члены команды должны **взаимодействовать**



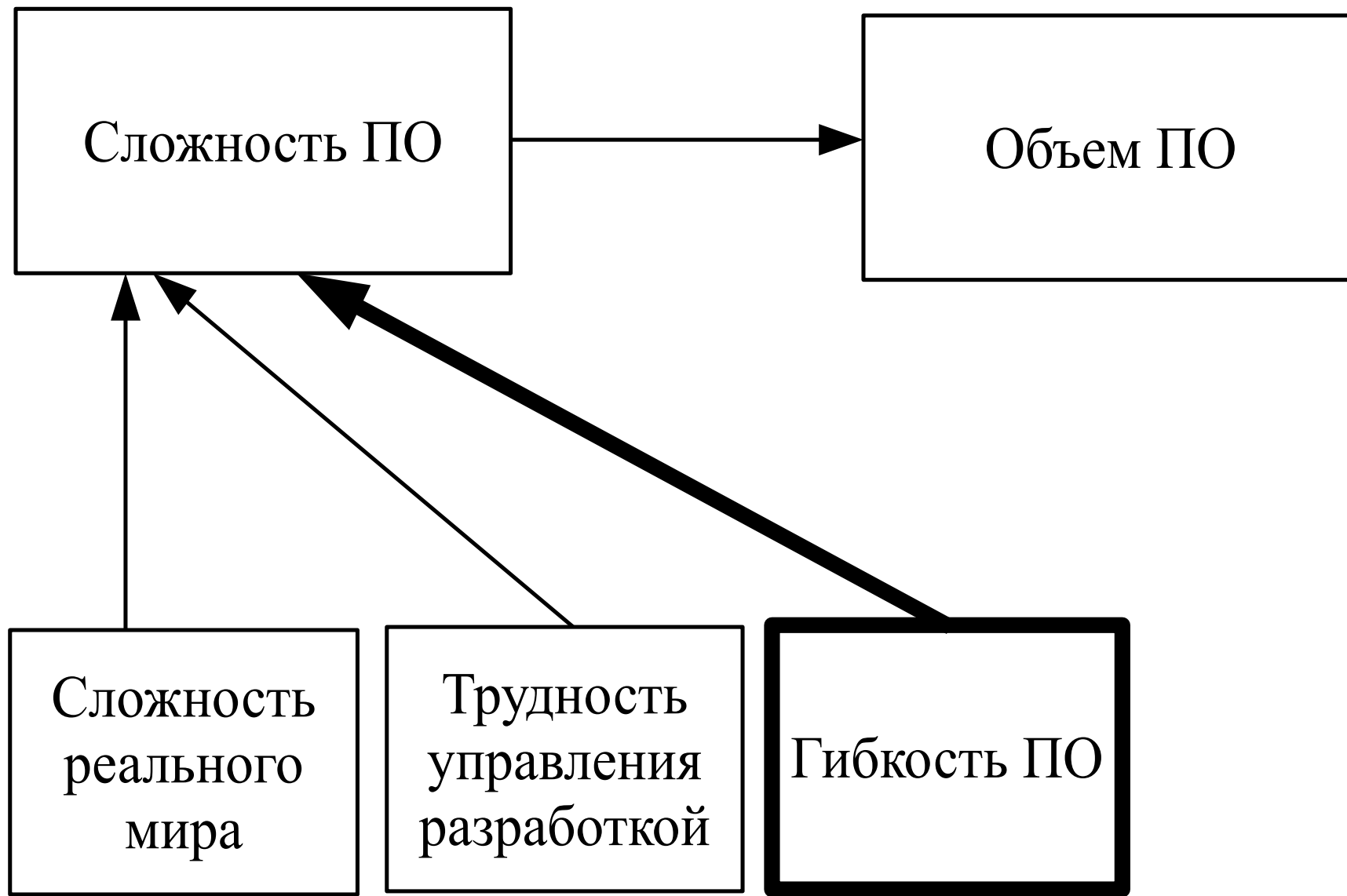
- ✓ Члены команды должны **взаимодействовать**
- ✓ **Закон Брукса:** добавление разработчиков увеличивает сроки реализации проекта — время для понимания сути проекта + увеличение количества связей в команде



- ✓ Члены команды должны **взаимодействовать**
- ✓ **Закон Брукса**: добавление разработчиков увеличивает сроки реализации проекта — время для понимания сути проекта + увеличение количества связей в команде
- ✓ Затраты на **системное тестирование** есть функция от числа разработчиков

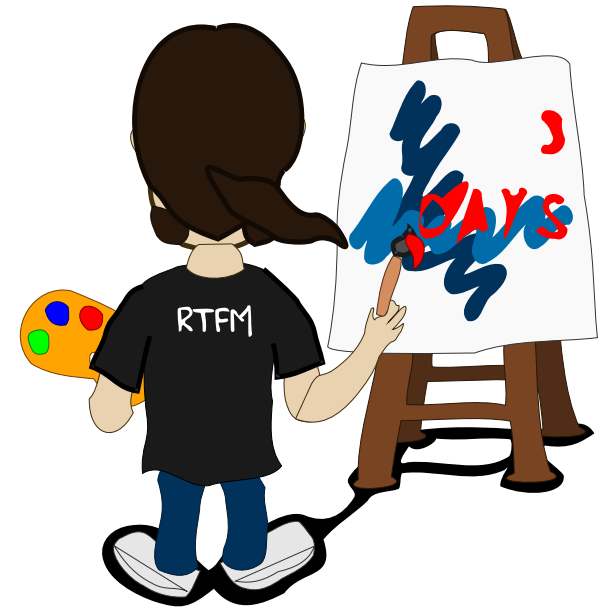


# Причины сложности





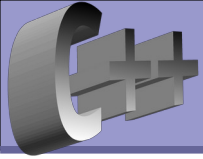
- ✓ Разработка ПО как **творческий процесс** стимулирует создание «лучшего из всех» с чистого листа



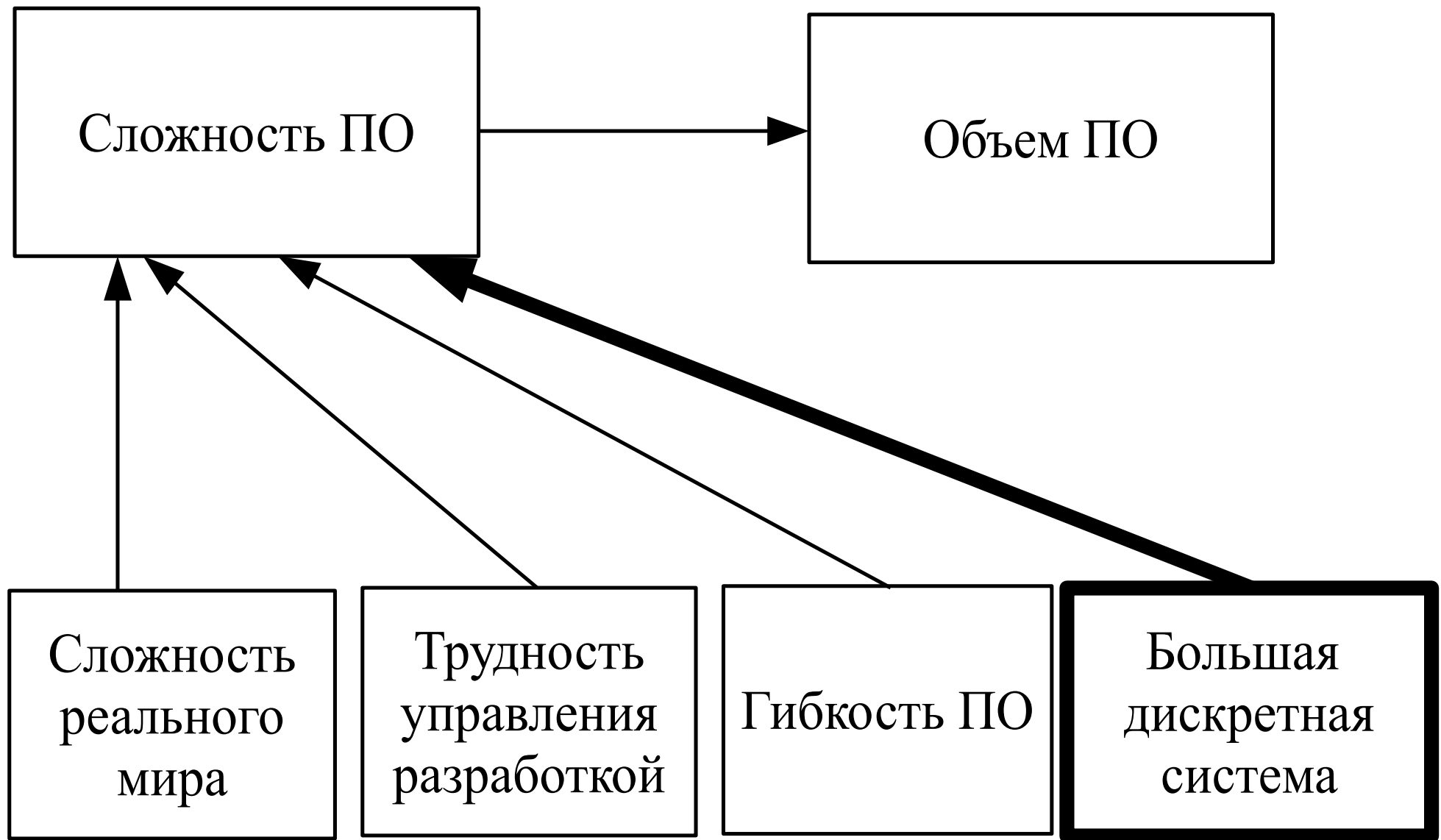
- ✓ Разработка ПО как **творческий процесс** стимулирует создание «лучшего из всех» с чистого листа
- ✓ Все программисты — **ОПТИМИСТЫ**



- ✓ Разработка ПО как **творческий процесс** стимулирует создание «лучшего из всех» с чистого листа
- ✓ Все программисты — **ОПТИМИСТЫ**
- ✓ **Отсутствие единых стандартов** поощряет на разработку своих базовых блоков (для строительства здания возводим кирпичные заводы)



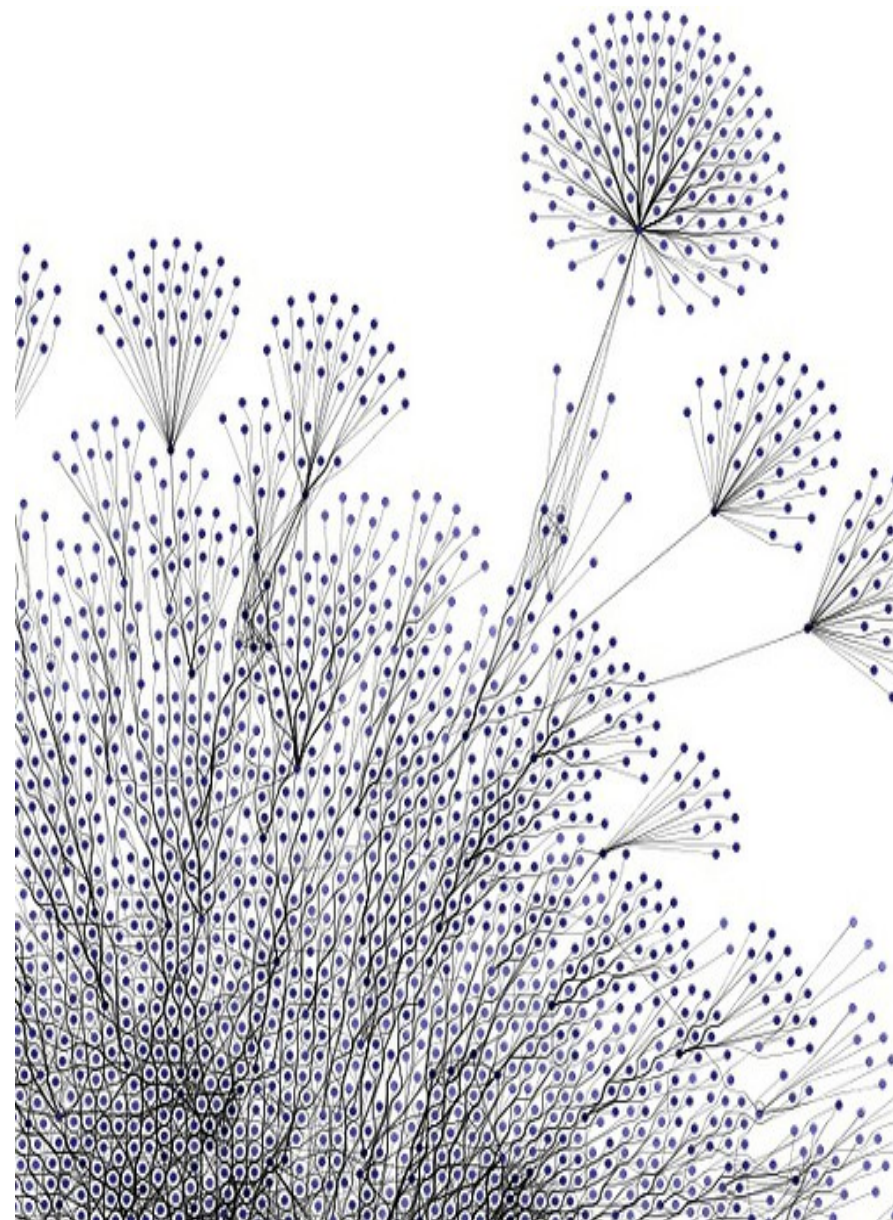
# Причины сложности





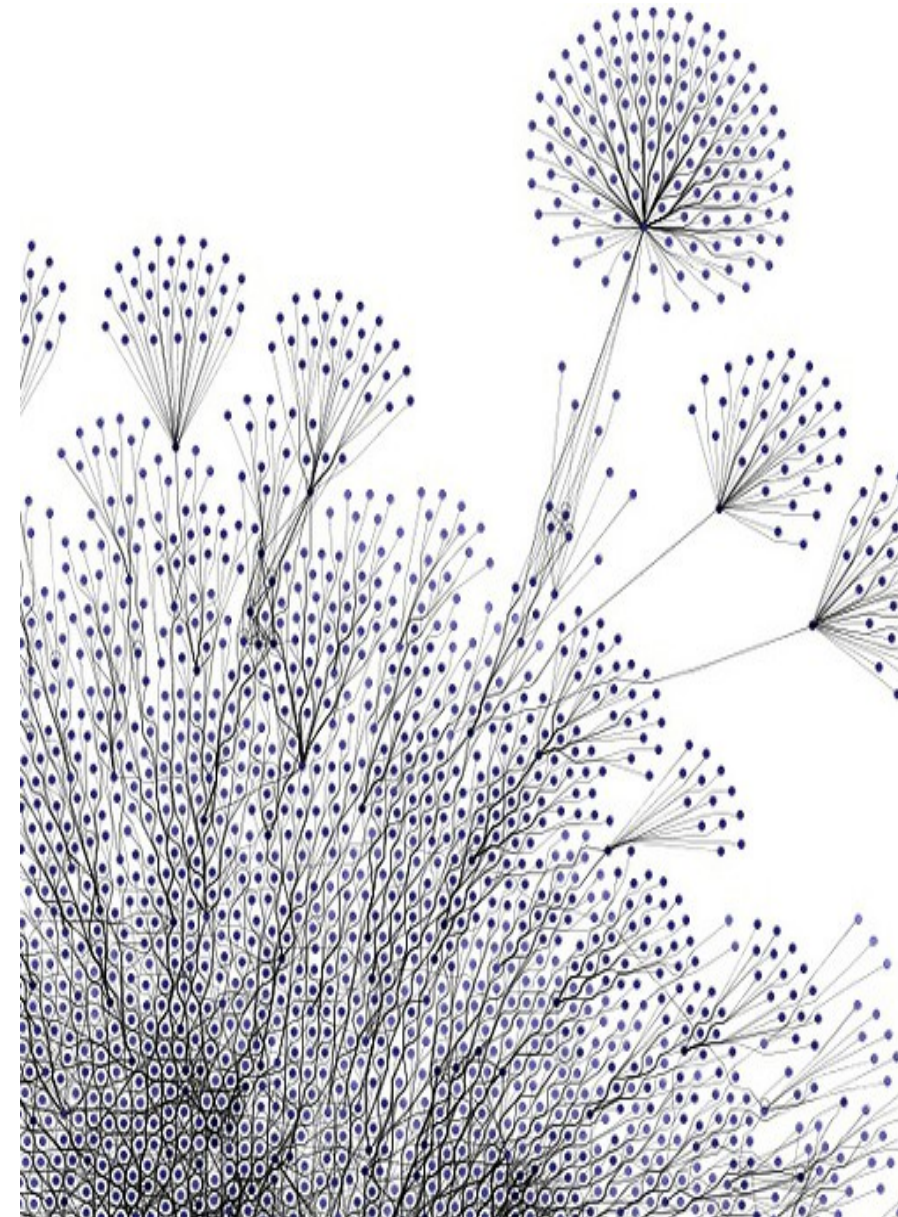


# Большая дискретная система





- ✓ Огромное количество возможных **состояний** даже в простейших программах



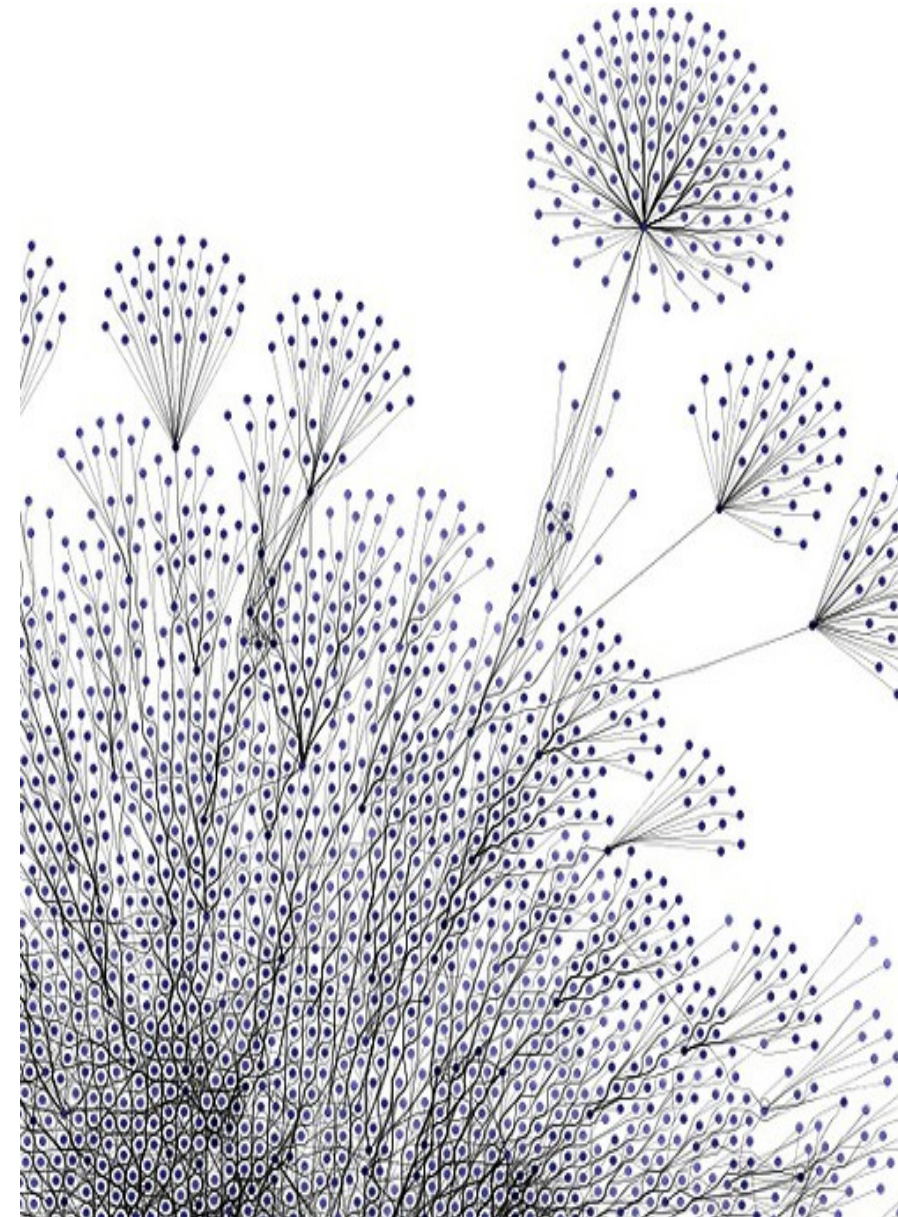




# Большая дискретная система



- ✓ Огромное количество возможных **состояний** даже в простейших программах
- ✓ Немыслимое количество возможных **переходов** из состояния в состояние

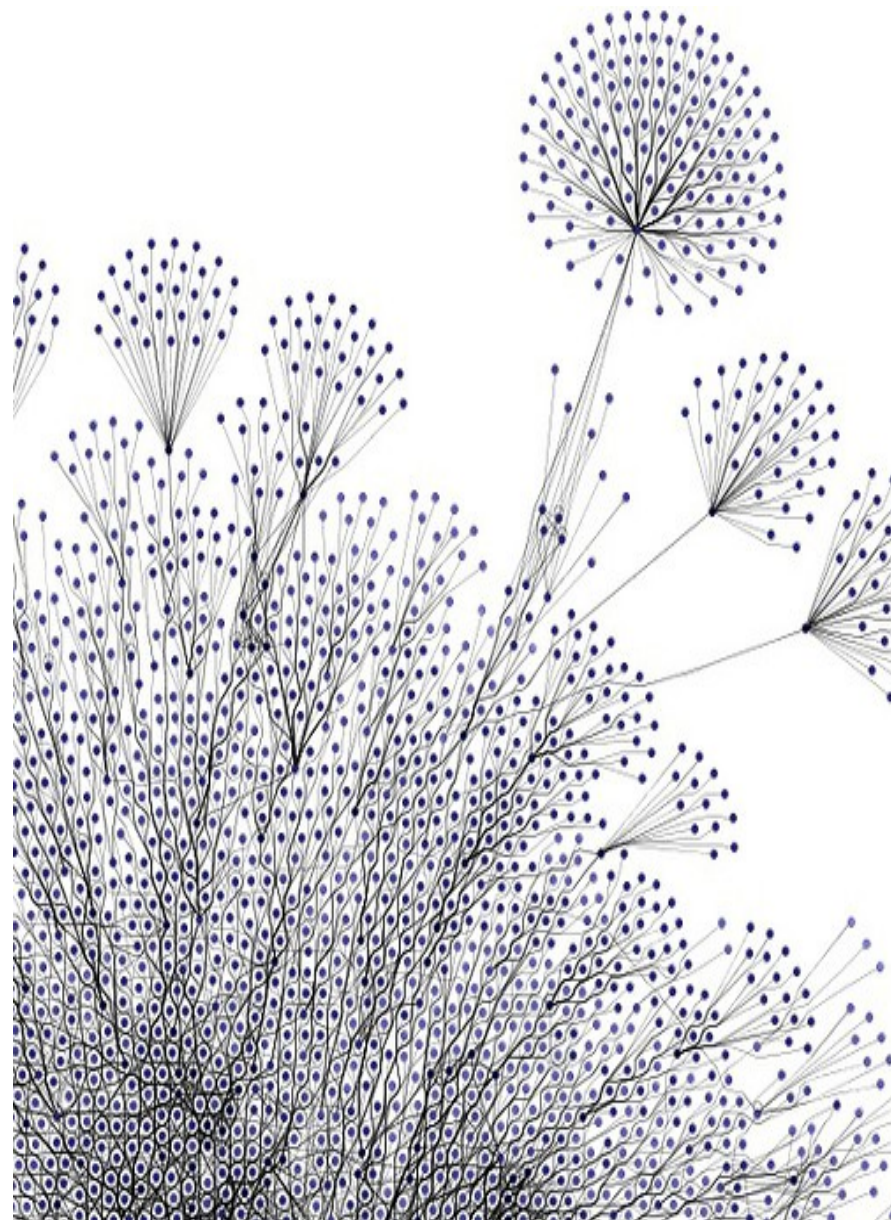




# Большая дискретная система



- ✓ Огромное количество возможных **состояний** даже в простейших программах
- ✓ Немыслимое количество возможных **переходов** из состояния в состояние
- ✓ Объективные трудности в **математическом аппарате** исследования больших дискретных систем



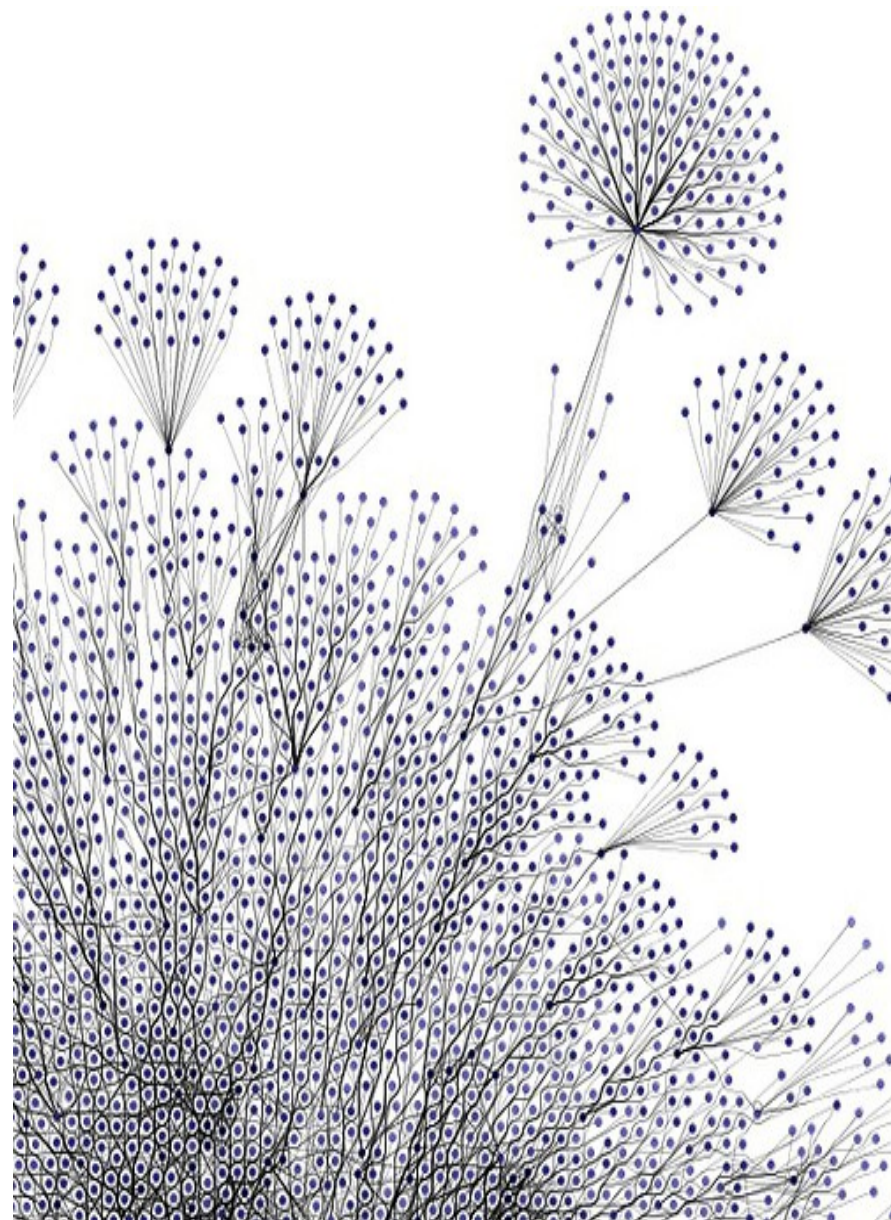




# Большая дискретная система



- ✓ Огромное количество возможных **состояний** даже в простейших программах
- ✓ Немыслимое количество возможных **переходов** из состояния в состояние
- ✓ Объективные трудности в **математическом аппарате** исследования больших дискретных систем
- ✓ Никогда нельзя быть уверенным в **безошибочности** программы





# **Введение в системное программирование**