

Технология разработки программного обеспечения

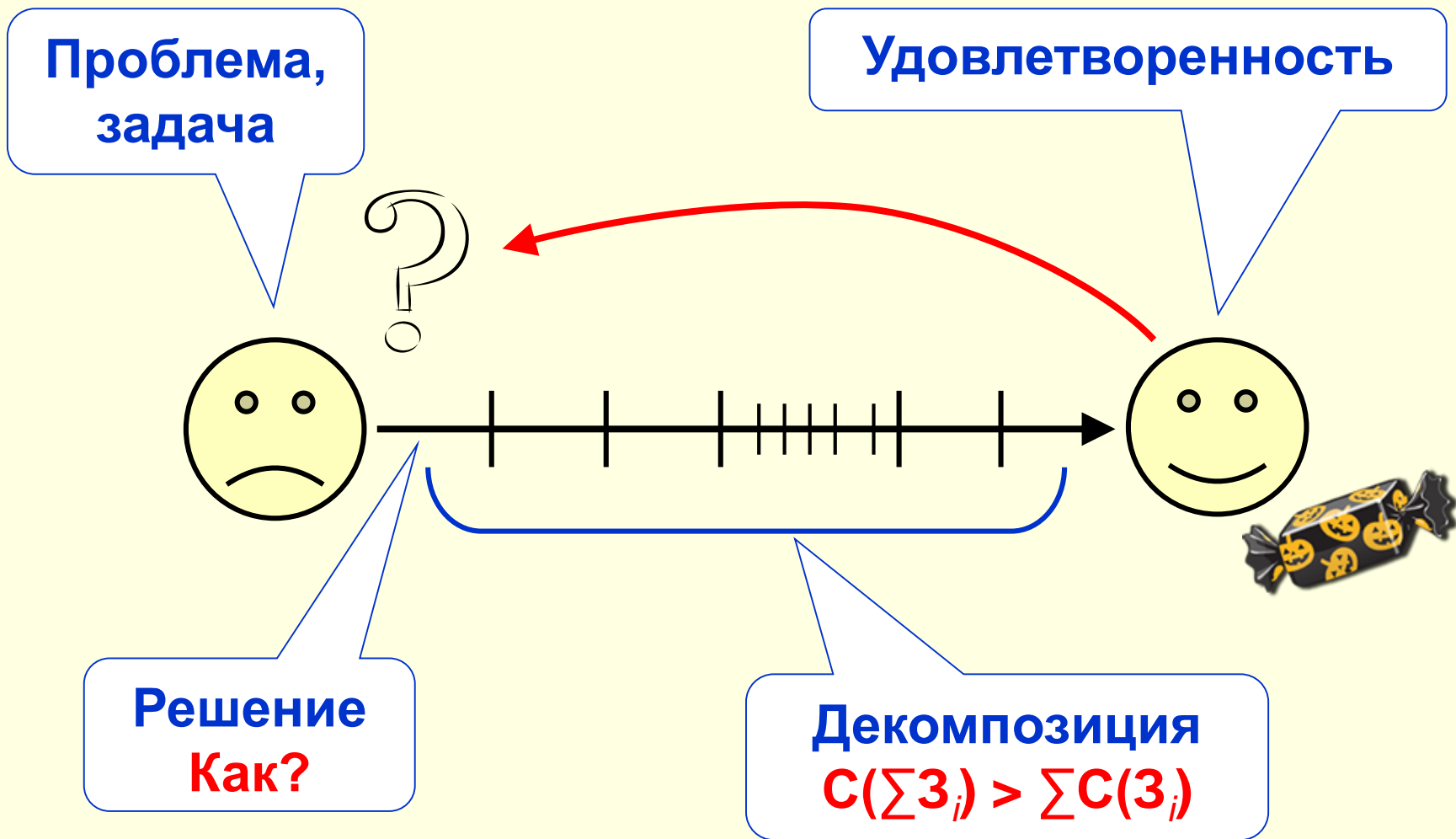
1

- Введение
- Жизненный цикл программного обеспечения

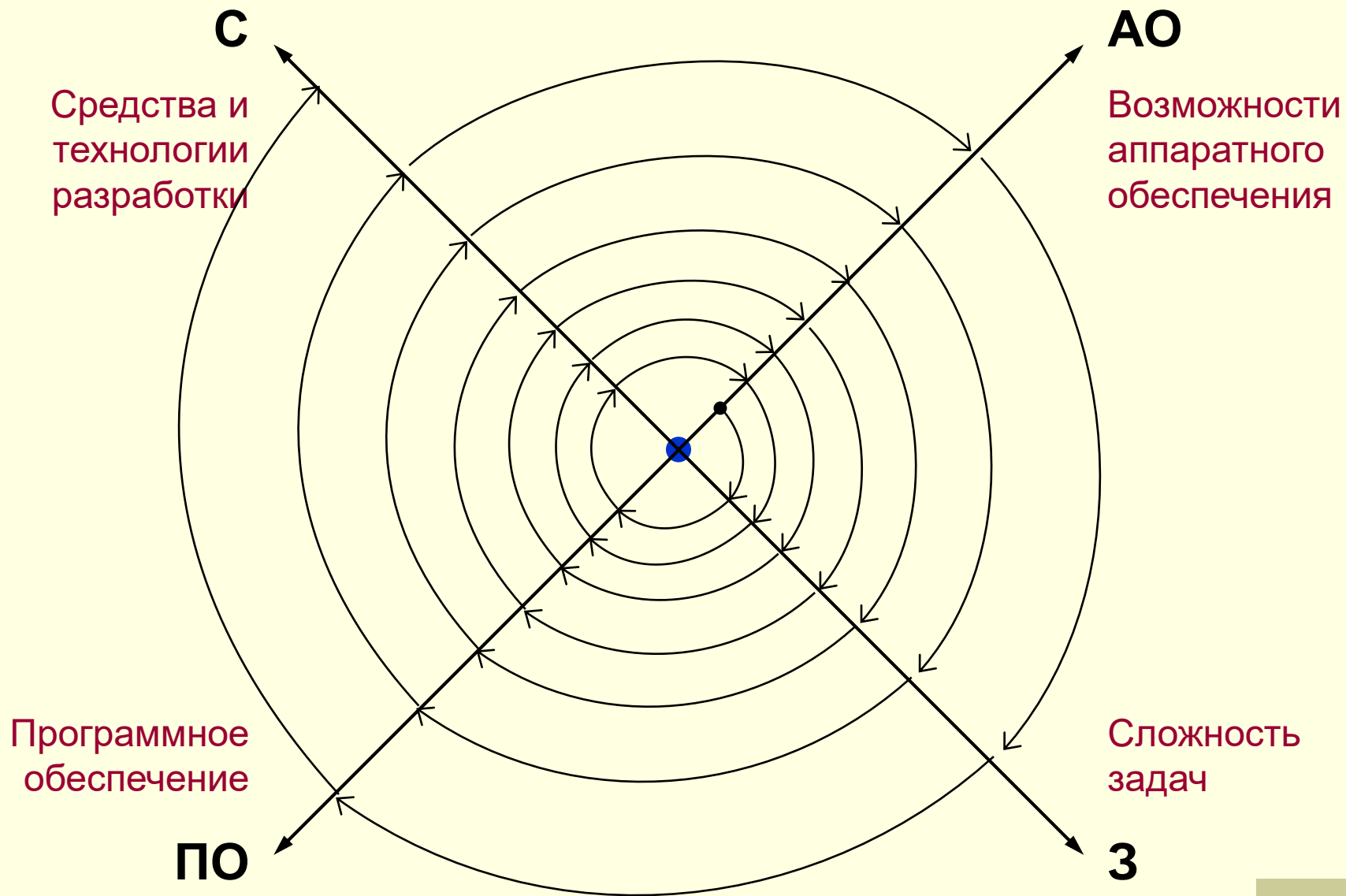
Гришмановский Павел Валерьевич

доцент кафедры автоматики и компьютерных систем,
к.т.н., доцент

Мета



Мета



Мета

Поколения языков программирования

1. машинно-ориентированные
2. с мнемоническим представлением команд (ассемблеры)
3. высокого уровня
4. прикладного программирования и среды разработки
5. визуального программирования

Группы языков программирования по назначению, предметной области и классам задач

- численных расчетов
- обработки деловой информации
- искусственного интеллекта
- системного программирования
- систем управления
- общего назначения

Парадигмы программирования

- императивная (процедурная)
- аппликативная (функциональная)
- логическая
- объектно-ориентированная

Парадигма – совокупность фундаментальных установок, представлений и терминов, принимаемая и разделяемая сообществом

Парадигма программирования –

- 1) система идей и понятий, определяющих стиль написания программ и образ мышления программиста
- 2) способ концептуализации, определяющий организацию вычислений и структурирование работы, выполняемой компьютером

Теоретический курс

- Жизненный цикл и процесс разработки ПО
- Методы анализа и проектирования ПО
- Средства моделирования ПО
- Средства разработки ПО и поддержки процесса

Практический курс

- Освоение среды и технологии разработки ПО
- Выполнение проекта по разработке продукта

Основная литература

1. Орлов С.А. Технологии разработки программного обеспечения. Учебник. – СПб.: Питер, 2012.
2. Гагарина Л.Г., Кокорева Е.В., Виснадул Б.Д. Технологии разработки программного обеспечения: учебное пособие – М.: ИД «Форум»: Инфра-М, 2017.
3. Брауде Э. Технология разработки программного обеспечения. – СПб.: Питер, 2004.
4. Рудаков А.В. Технология разработки программных продуктов : учеб. пособие для студ. сред. проф. образования. – М.: Издательский центр «Академия», 2006.
5. Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. – СПб.: Питер, 2002.
6. Буч Г., Рамбо Дж., Джекобсон А. Язык UML. Руководство пользователя. – М.: ДМК, 2006.
7. Бек К. Экстремальное программирование: библиотека программиста. – СПб.: Питер, 2018.
8. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++. – М.: Вильямс, 2008.

Введение

Технология – комплекс организационных мер, операций и приемов, направленных на изготовление, обслуживание, ремонт, эксплуатацию и/или утилизацию изделия (предмета технологии) с **номинальным качеством и оптимальными затратами**, и обусловленных текущим уровнем развития науки, техники и общества в целом

- **изделие** – любой конечный продукт труда (материальный, интеллектуальный и др.)
- **номинальное качество** – прогнозируемое или заранее заданное
- **оптимальные затраты** – минимально возможные, не влекущие ухудшение условий труда, нарушения норм, финансовых, экономических и пр. рисков

искусство, мастерство, умение + мысль, причина, наука
= методика, способ производства

Жизненный цикл технологии – совокупность стадий от зарождения технологических нововведений до их рутинизации

Введение

Стадии технологии	Потребители технологии
Новейшая – любая новая технология с высоким потенциалом	Новаторы – следят за технологическими новинками и стараются получить к ним доступ до широкого появления их на рынке
Передовая – зарекомендовала себя, но имеет пока небольшое распространение на рынке	Ранние последователи – осознают степень соответствия между преимуществами новой технологии и их интересами
Современная – является стандартом, повышается спрос на эту технологию	Раннее большинство – важным фактором является практическое значение
Не новая – полезная технология, но уже существует более новая, вызывающая падение спроса	Позднее большинство – ждут, когда продукт станет общепризнанным стандартом
Устаревшая – заменяется более совершенной, происходит отказ в пользу новой технологии	Отстающие – не следят за технологическими новинками и приобретают, если эта технология внедрена в какой-либо товар, но не пользуются ей

Введение

Технология разработки (конструирования) **программного обеспечения** (ТРПО) – система инженерных принципов для создания экономичного программного обеспечения (ПО), которое надежно и эффективно работает в реальных компьютерах

методы + средства + процедуры
= технология

Технология разработки программного обеспечения направлена на промышленную разработку ПО соответствующего качества, в установленные сроки и при установленных затратах

Введение

Методы обеспечивают решение задач

- анализ системных и программных требований
- планирование и оценка проекта
- проектирование алгоритмов, структур данных и программных структур
- кодирование, отладка, тестирование
- документирование и сопровождение

Средства обеспечивают автоматизированную или автоматическую поддержку методов

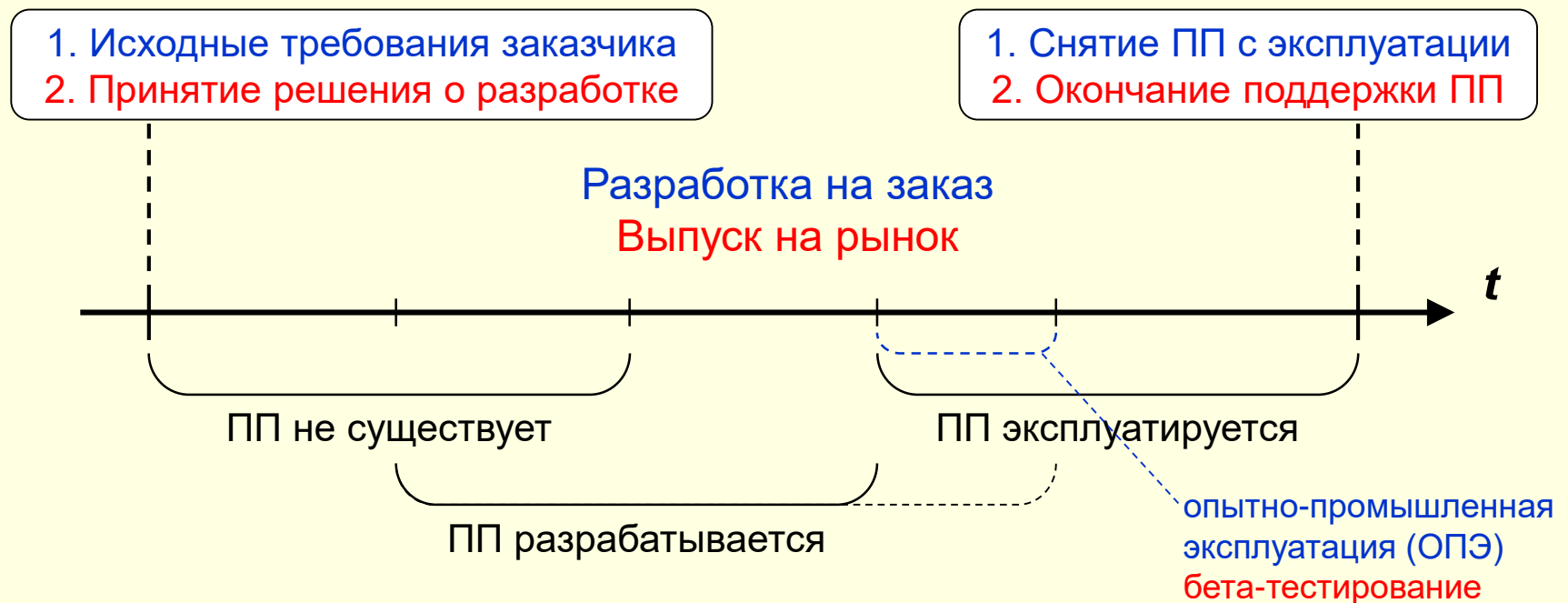
Процедуры соединяют методы и средства – обеспечивают непрерывную технологическую цепочку разработки

- порядок применения методов и средств
- формирование отчетов, форм по соответствующим требованиям
- контроль и оценка, необходимые для обеспечения качества и изменений
- формирование вех, по которым осуществляется оценка прогресса

Процесс разработки ПО – *последовательность* шагов, использующих методы, средства и процедуры

Жизненный цикл ПО

Жизненный цикл программного обеспечения – интервал времени от момента принятия решения о разработке программного продукта до момента снятия его с эксплуатации

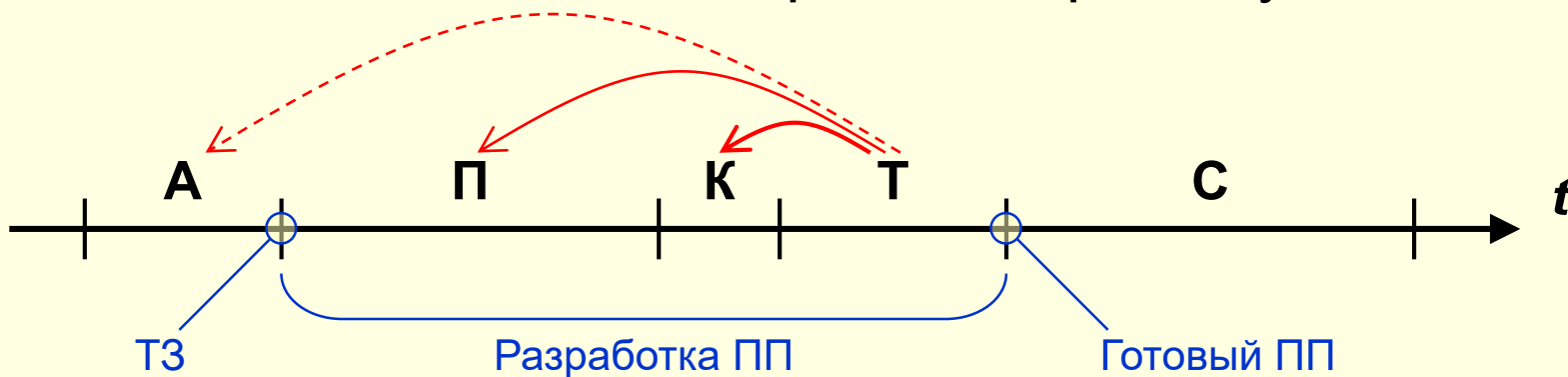


Жизненный цикл ПО



Модели жизненного цикла ПО

Классическая (каскадная) модель – очередной этап начинается только после завершения предыдущего



Достоинства

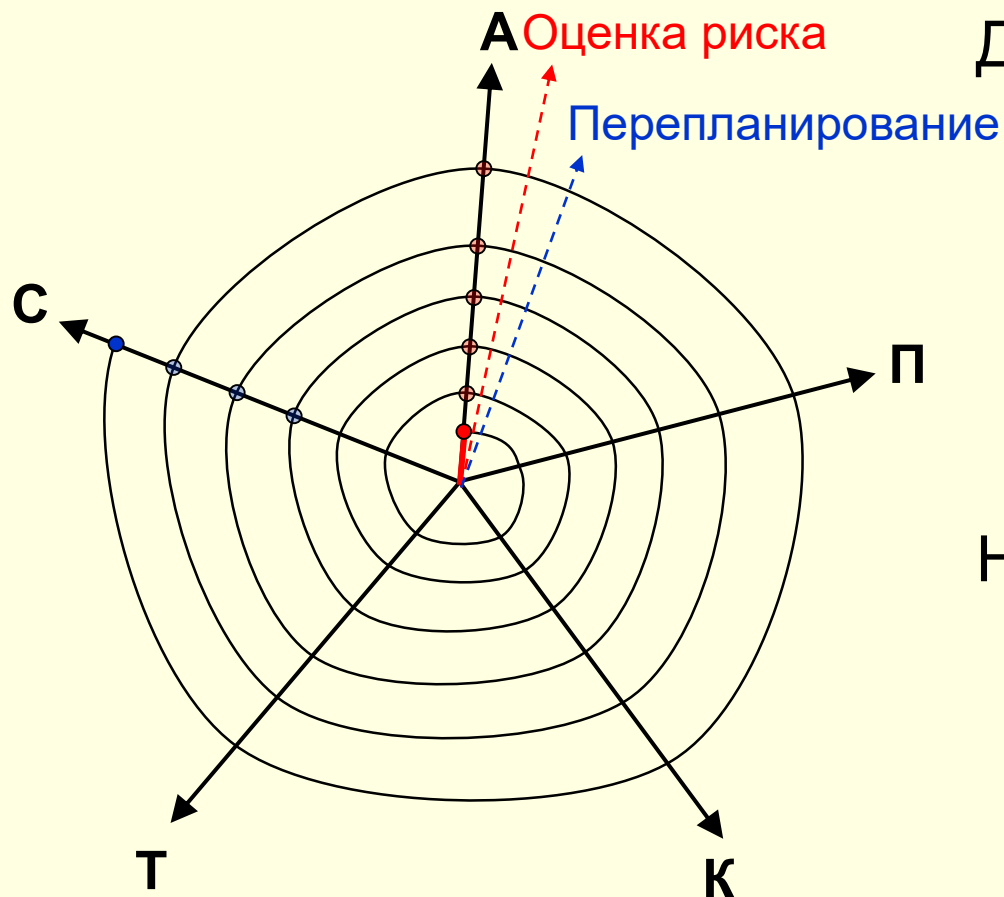
- хорошая планируемость и контролируемость процесса разработки

Недостатки

- в начале разработки должны быть известны все требования
- результаты доступны заказчику только в конце разработки
- неустойчивость к изменениям и ошибкам – затягивание сроков

Модели жизненного цикла ПО

Спиральная (CASE) модель – фрагменты каждого из этапов выполняются последовательно на каждой итерации



Достоинства

- позволяет учитывать риск на каждой итерации
- устойчивость к изменениям и ошибкам
- возможны промежуточные версии

Недостатки

- повышенные требования к заказчику
- трудности планирования и контроля разработки

Стратегии разработки ПО

- **Стратегия однократного прохода** – линейная последовательность этапов (соответствует классической модели ЖЦПО)
- **Эволюционная стратегия** – ПП строится в виде последовательности версий, но в начале требования определены не полностью и уточняются в процессе разработки (соответствует спиральной модели ЖЦПО)
- **Инкрементная стратегия** – ПП строится в виде последовательности версий, но в начале требования полностью определены, оставшаяся часть разработки выполняется в виде последовательности версий, каждая реализует запланированную часть возможностей, (сочетает качества классической и спиральной моделей ЖЦПО)

Стратегии разработки ПО

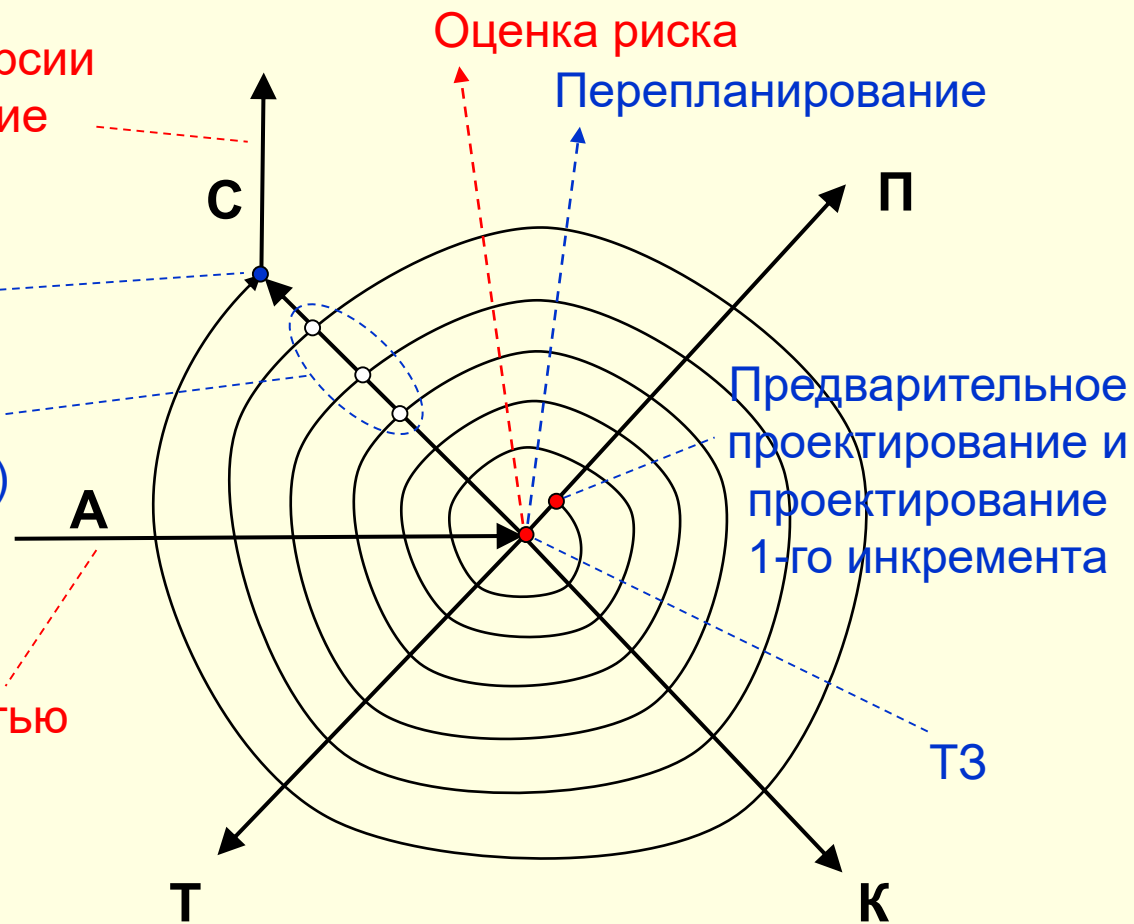
Инкрементная стратегия

После выпуска финальной версии
продолжается сопровождение
(как в каскадной модели)

Финальная версия

Промежуточные версии
(как в спиральной модели)

Анализ выполняется полностью
до начала разработки
(как в каскадной модели)



Стратегии разработки ПО

Параметр	Стратегия		
	Однократный проход	Инкрементная	Эволюционная
Требования полностью определены в начале	Да	Да	Нет
Промежуточные версии ПП	Нет	Возможны	Да
Итерационный процесс	Нет	Да	Да
Выполнение анализа риска	Однократно вначале	Возможен на каждой итерации	На каждой итерации
Условия разработки	На заказ	На заказ На рынок	На рынок

Виды процессов

Прогнозирующий процесс (тяжеловесный)

- Детальное планирование и тотальный контроль
- Тотальное документирование и архивирование
- Большой коллектив специалистов средней и низкой квалификации
- Не учитываются человеческий фактор и личные качества
- Стратегия однократного прохода или инкрементная

Адаптивный процесс (облегченный)

- Общее планирование и контроль по вехам
- Только необходимая проектная документация
- Небольшой коллектив специалистов высокой квалификации
- Эффективное использование личных качеств исполнителей
- Инкрементная или эволюционная стратегии

XP – экстремальное программирование

XP – это экстремум **адаптивного процесса**

1. Игра планирования – постоянно
2. Частая смена версий – итерация не более 2-х недель
3. Метафора – общедоступная история об устройстве и функционировании системы
4. Простое проектирование – наиболее простая идея, которая будет работать
5. Тестирование – «Тестируй, а затем кодируй»
6. Реорганизация – улучшать систему без изменения ее функционирования
7. Парное программирование – постоянная верификация
8. Коллективное владение кодом – абсолютная доступность всего кода всем участникам
9. Непрерывная интеграция – непрерывное регрессионное тестирование по мере завершения каждой задачи
10. 40-часовая неделя – недопустимы сверхурочные работы
11. Локальный заказчик – уточнение требований и оценка прогресса
12. Стандарты кодирования – единые правила и стиль