



Новокузнецкий  
институт (филиал)  
Кемеровский  
государственный  
университет

— 1939

# Процесс создания программного обеспечения

Языки и методы программирования

# План лекции

1. Жизненный цикл программного обеспечения:
  - процессы жизненного цикла;
  - модель жизненного цикла.
2. Инструментарий технологии программирования.
3. Методологии разработки программного обеспечения.
  - прогнозируемые (предикативные) методологии;
  - Адаптивные (гибкие) методологии.



Новокузнецкий  
институт (филиал)  
Кемеровский  
государственный  
университет

— 1939

# Жизненный цикл программного обеспечения

# Жизненный цикл программного обеспечения

**Жизненный цикл программного обеспечения** – это период времени, который начинается с момента принятия решения о необходимости создания ПО и заканчивается в момент его полного изъятия из эксплуатации.

Основной нормативный документ, регламентирующий состав процессов ЖЦ ПО – это международный стандарт ISO/IEC 12207: 1995 «Information Technology -Software Life Cycle Processes». Определяет структуру ЖЦ, содержащую процессы действия и задачи, которые должны быть выполнены во время создания ПО.

**Программный продукт** – это набор компьютерных программ, процедур и, возможно, связанной с ними документации и данных.

**Процесс** – это совокупность взаимосвязанных действий, преобразующих некоторые входные данные в выходные.

# Процессы жизненного цикла программного обеспечения

## Основные процессы

Приобретение

Поставка

Разработка

Эксплуатация

Сопровождение

## Организационные процессы

Управление

Усовершенствование

Создание инфраструктуры

Обучение

## Вспомогательные процессы

Документирование

Верификация

Аудит

Аттестация

Разрешение проблем

Обеспечение качества

Совместная оценка

Управление конфигурацией

# Приобретение и поставка программного средства

Инициирование приобретения

Подготовка заявочных  
предложений

Подготовка и корректировка  
договора

Надзор за деятельностью  
поставщика

Приемка и завершение работ

Инициирование поставки

Подготовка ответа на  
заявочные предложения

Подготовка договора

Планирование работ по  
договору

Выполнение и контроль  
договорных работ и их оценка

Поставка и завершение работ

# Разработка программного средства

1. подготовительная работа;
2. анализ требований, предъявляемых к системе;
3. проектирование архитектуры системы;
4. анализ требований, предъявляемых к программному обеспечению;
5. проектирование архитектуры программного обеспечения;
6. детальное проектирование программного обеспечения;
7. кодирование и тестирование программного обеспечения;
8. интеграция программного обеспечения;
9. квалификационное тестирование программного обеспечения;
10. интеграция системы;
11. квалификационное тестирование системы;
12. установка программного обеспечения;
13. приемка программного обеспечения.

# Анализ требований

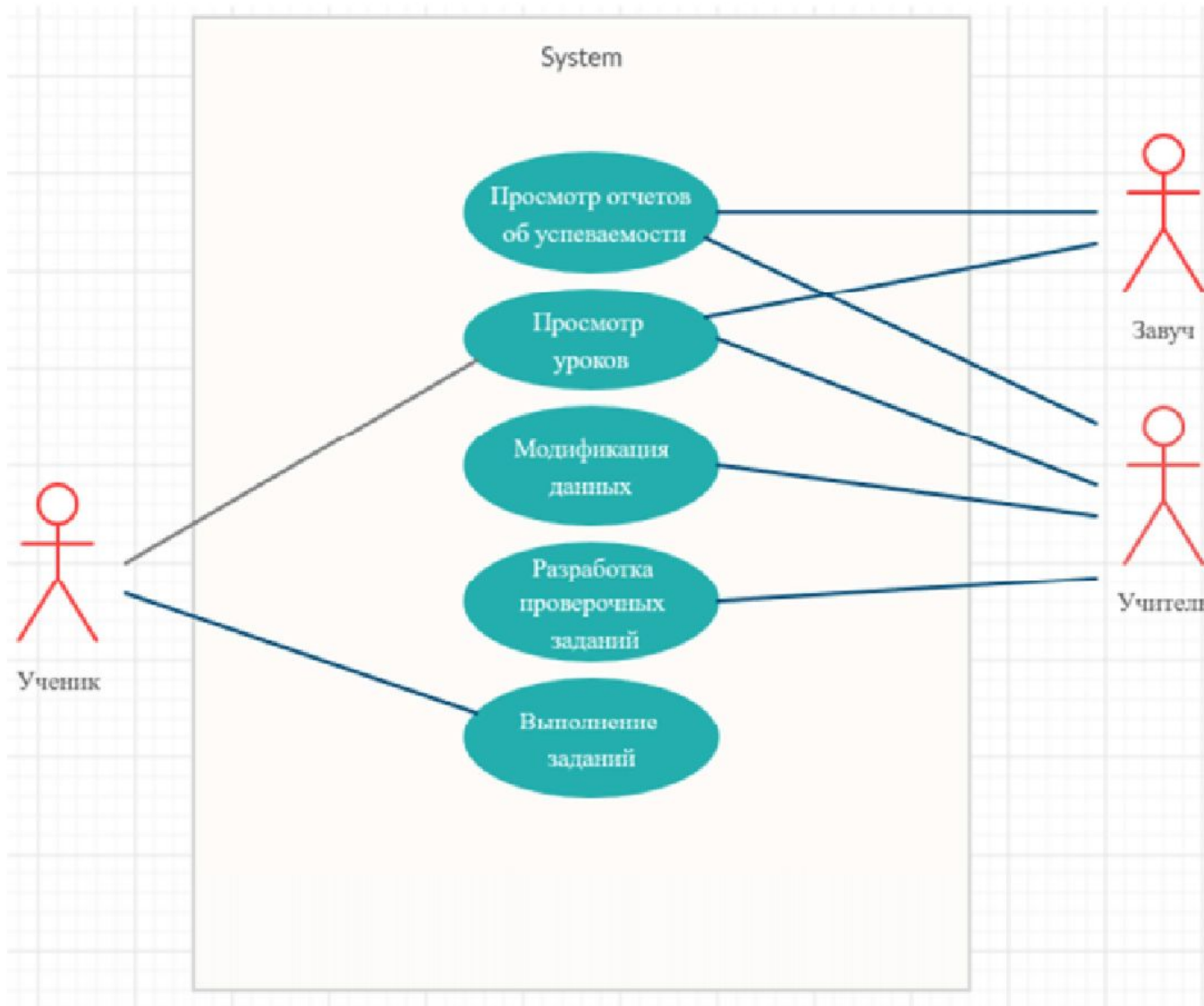
**Требование** – это некоторая функция, которая должна быть включена в создаваемую систему.

**Спецификация требований** – структурированный набор требований к программному обеспечению.

интервьюирование, анкетирование, мозговой штурм, сценарии и ролевые игры, создание прототипов, совместная разработка приложений, моделирование, uses case, наблюдение за деятельностью предприятия, изучение документов (особенно тех, которые используются для сбора или представления информации).



# Метод Uses Case



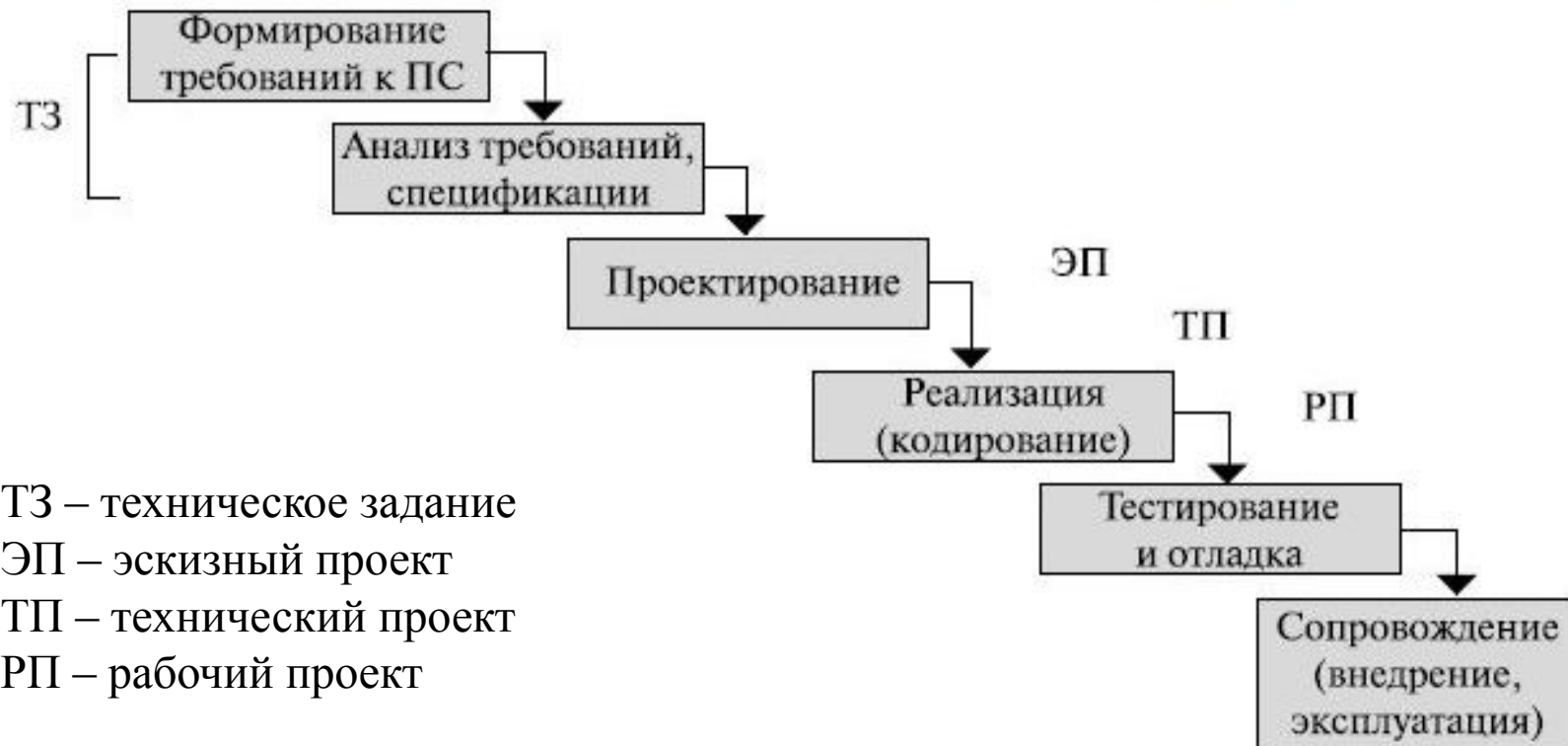
# Детальное проектирование ПО

- описание компонентов ПО и интерфейсов между ними на более низком уровне, достаточном для последующего кодирования и тестирования;
- разработка и документирование детального проекта базы данных;
- обновление (при необходимости) пользовательской документации;
- разработка и документирование требований к тестам и плана тестирования компонентов ПО;
- обновление плана интеграции ПО.

# Кодирование и тестирование ПО

- кодирование и документирование каждого компонента ПО, а также подготовка совокупности тестовых процедур и данных для их тестирования;
- тестирование каждого компонента ПО на соответствие предъявляемым к ним требованиям с последующим документированием результатов тестирования;
- обновление документации (при необходимости);
- обновление плана интеграции ПО.

# Модель жизненного цикла ПС



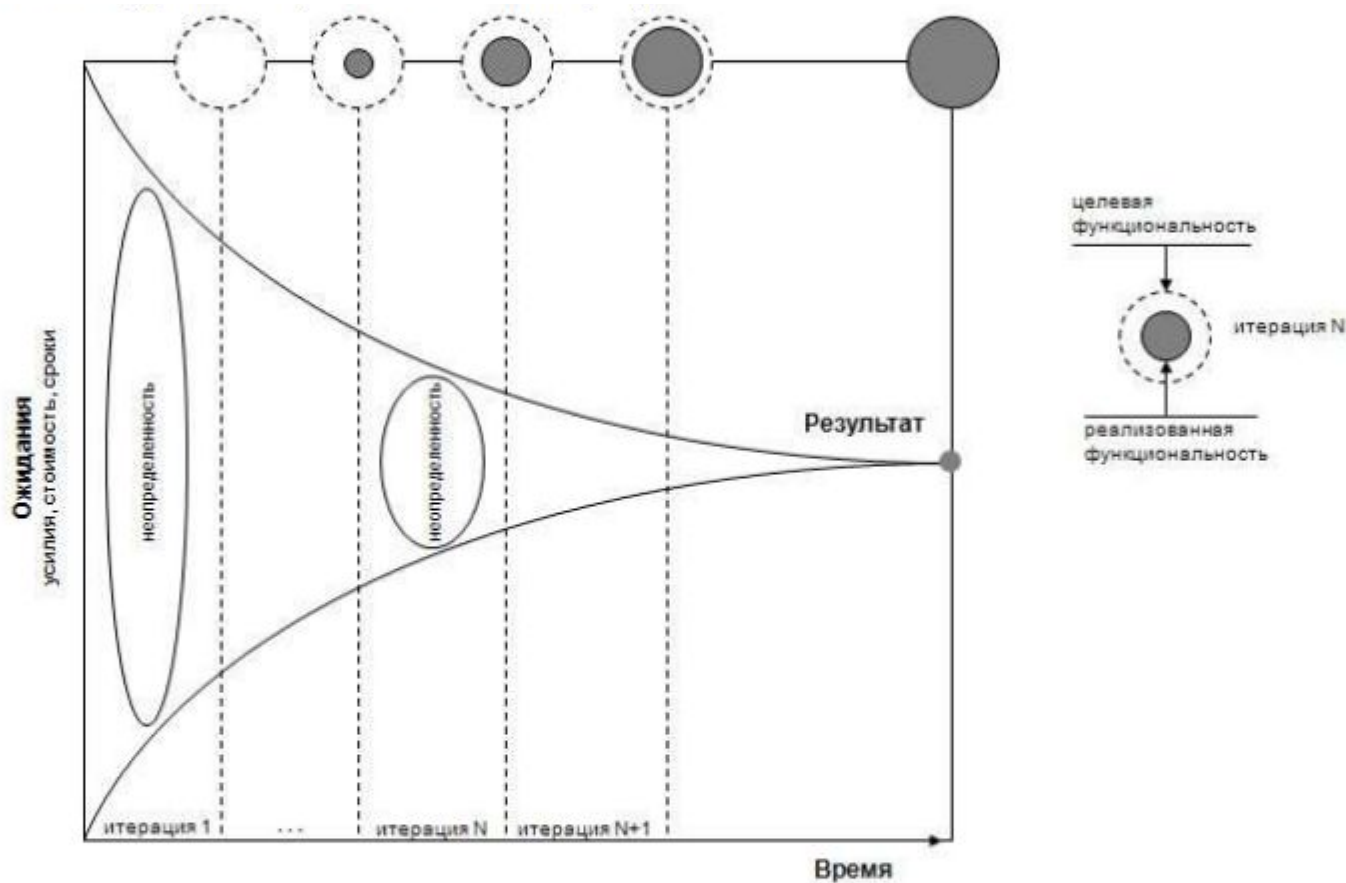
ТЗ – техническое задание

ЭП – эскизный проект

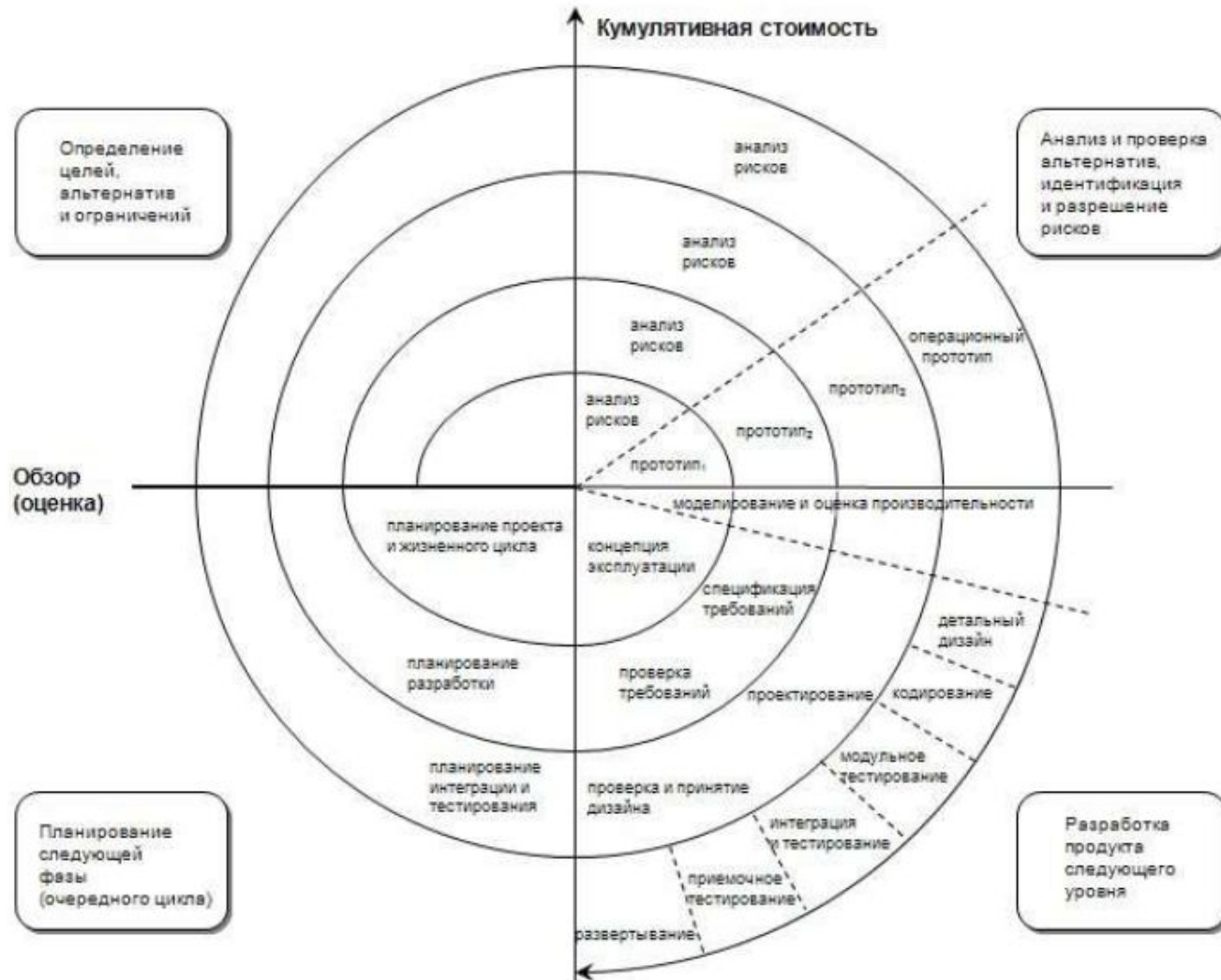
ТП – технический проект

РП – рабочий проект

# Оригинальная спиральная модель жизненного цикла разработки по Боэму



# Оригинальная спиральная модель жизненного цикла разработки по Боэму



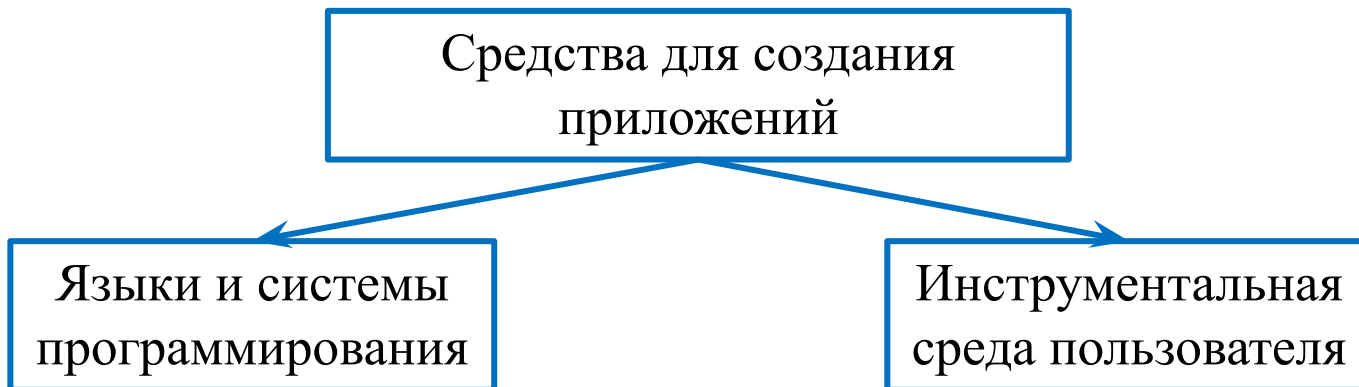


Новокузнецкий  
институт (филиал)  
Кемеровский  
государственный  
университет

— 1939

# Инструментарий технологии программирования

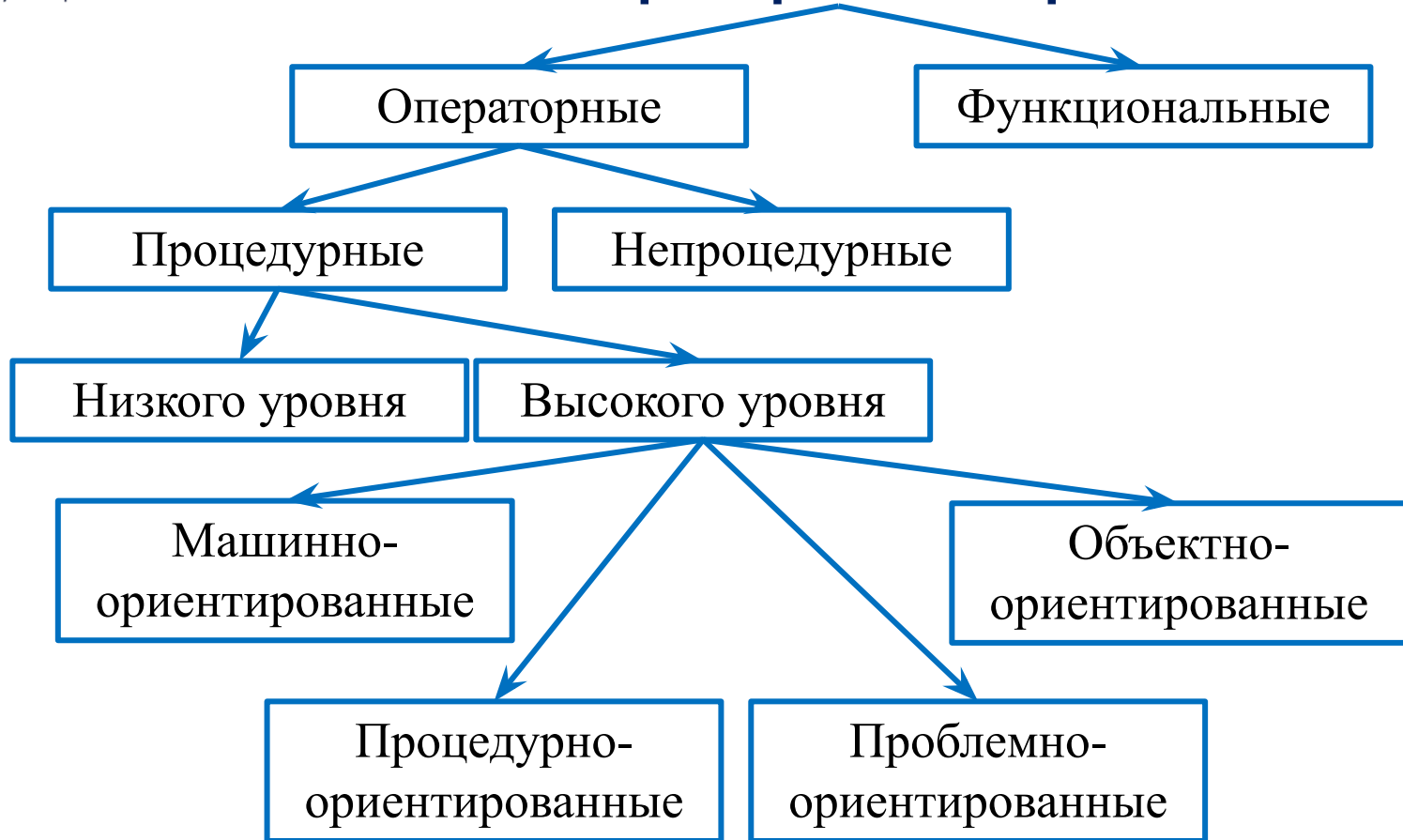
# Инструментарий технологии программирования



библиотека функций, процедур, объектов и методов обработки;  
макрокоманды;  
клавишные макросы;  
языковые макросы;  
конструкторы экранных форм и объектов;  
генераторы приложений;  
языки запросов высокого уровня;  
конструкторы меню

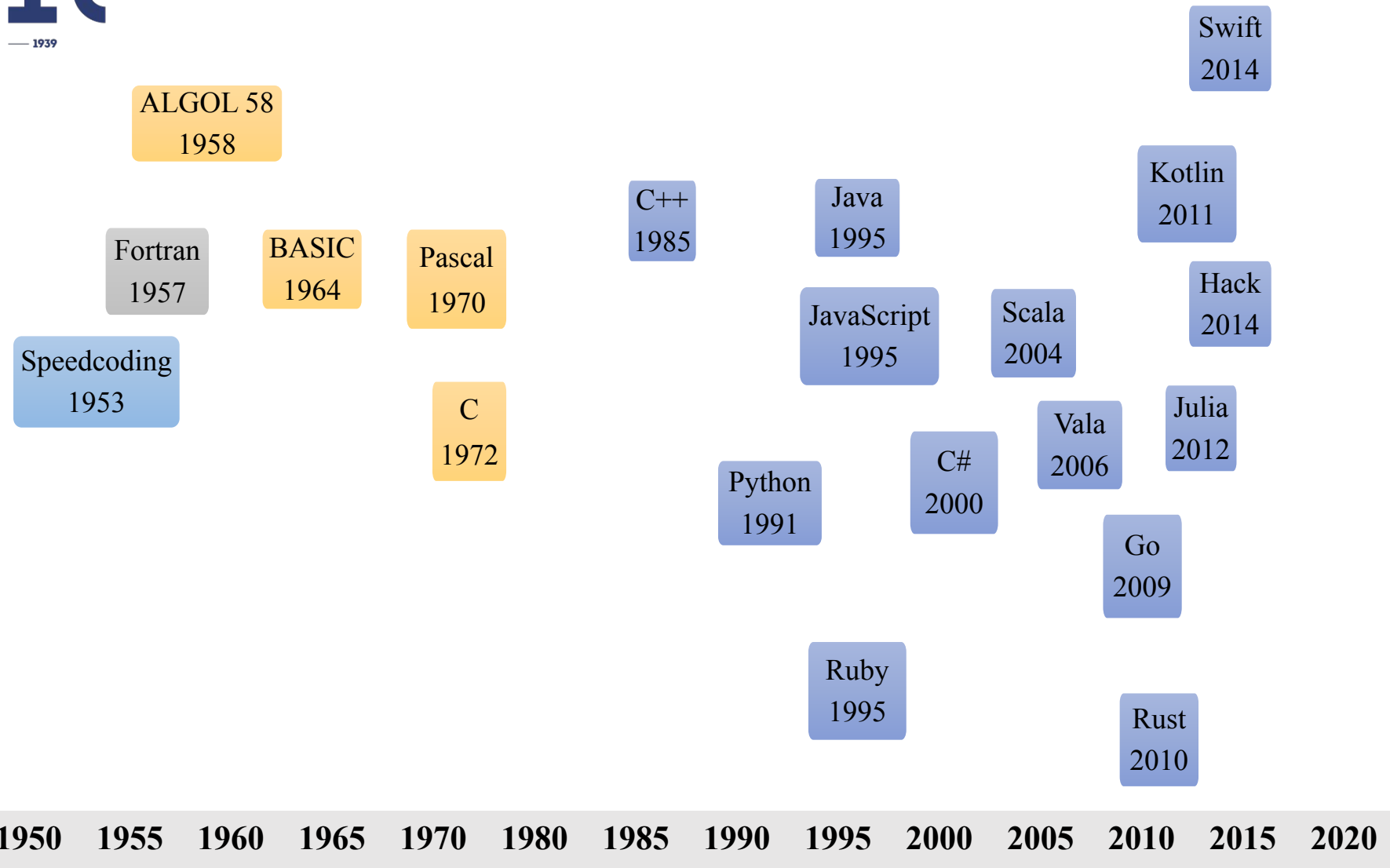


# Языки программирования





# Языки программирования





Новокузнецкий  
институт (филиал)  
Кемеровский  
государственный  
университет

— 1939

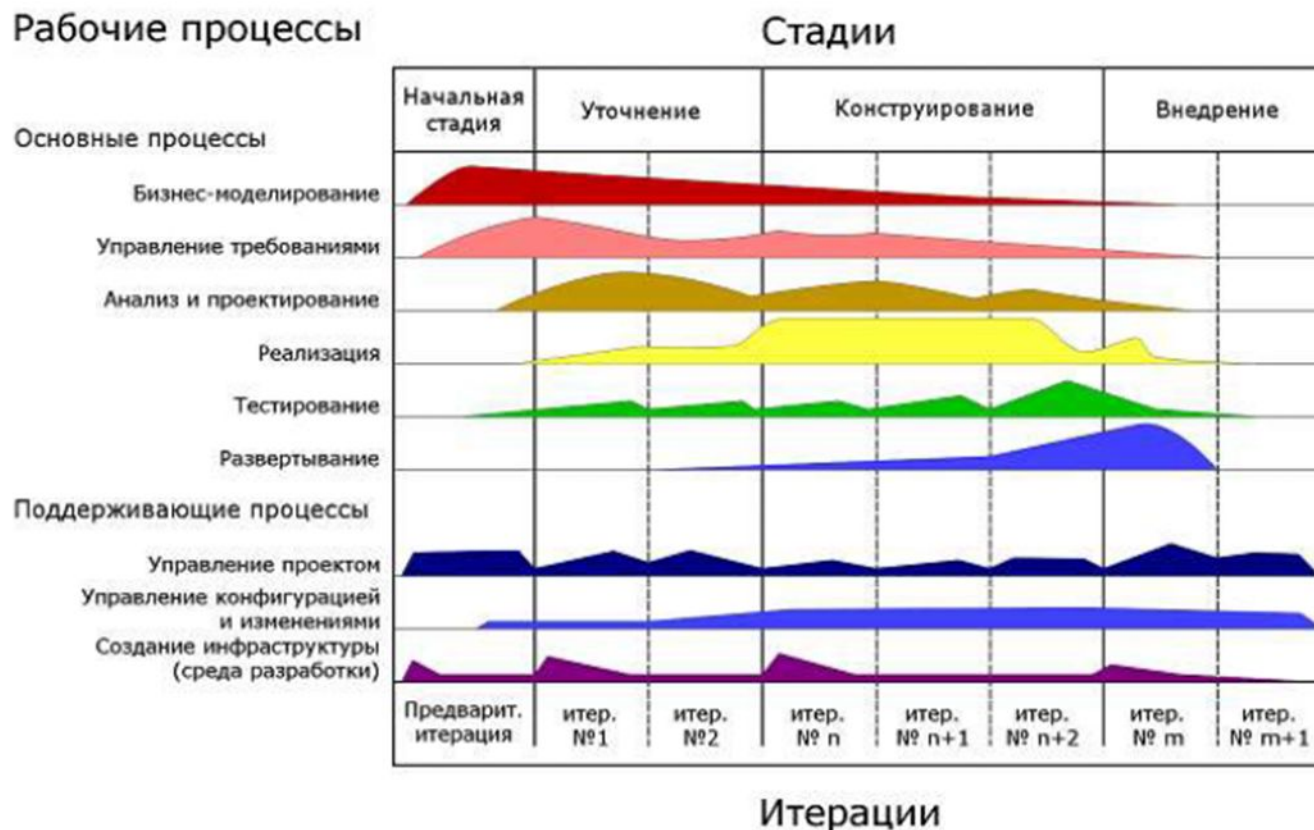
# Методологии разработки программного обеспечения

# Прогнозируемые (предикативные) методологии

- Детальное планирование будущего.
- Известны запланированные задачи и ресурсы на весь срок проекта.
- Команда с трудом реагирует на возможные изменения.
- План оптимизирован исходя из состава работ и существующих требований.
- Изменение требований может привести к существенному изменению плана, а также дизайна проекта

# RUP (Rational Unified Process)

Методология RUP описывает абстрактный общий процесс, на основе которого организация или проектная команда должна создать специализированный процесс, ориентированный на ее потребности.



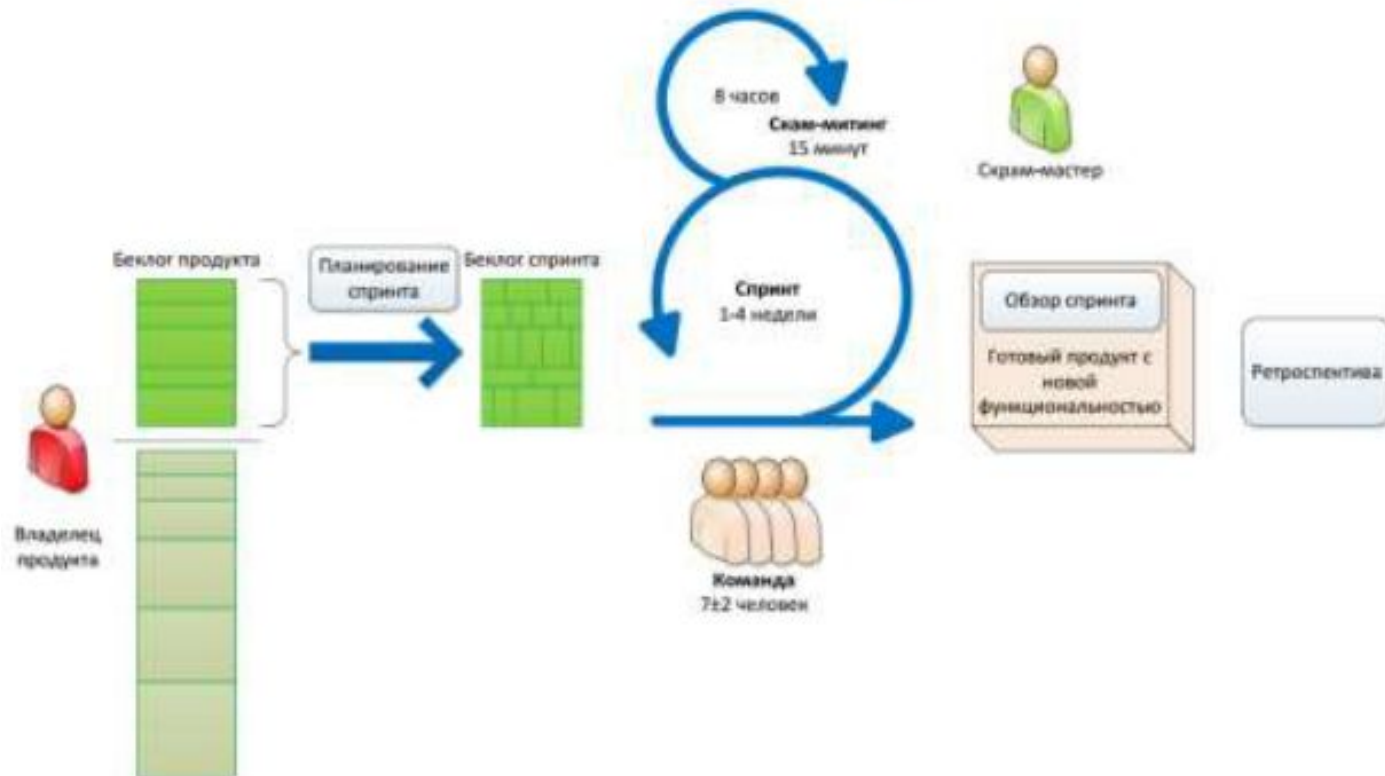
# Адаптивные (гибкие) методологии

Нацелены на преодоление ожидаемой неполноты требований и их постоянного изменения.

- Когда меняются требования, команда разработчиков тоже меняется.
- Команда, участвующая в адаптивной разработке, с трудом может предсказать будущее проекта.
- Существует точный план лишь на ближайшее время.
- Более удаленные во времени планы существуют лишь как декларации о целях проекта, ожидаемых затратах и результатах.

# Scrum

Scrum предоставляет эмпирический подход к разработке ПО. Этот процесс быстр, адаптивен, умеет подстраиваться и отличен от каскадной модели. Scrum основан на повторяющихся циклах, это делает его более гибким и предсказуемым.



# Экстремальное программирование (eXtreme Programming)

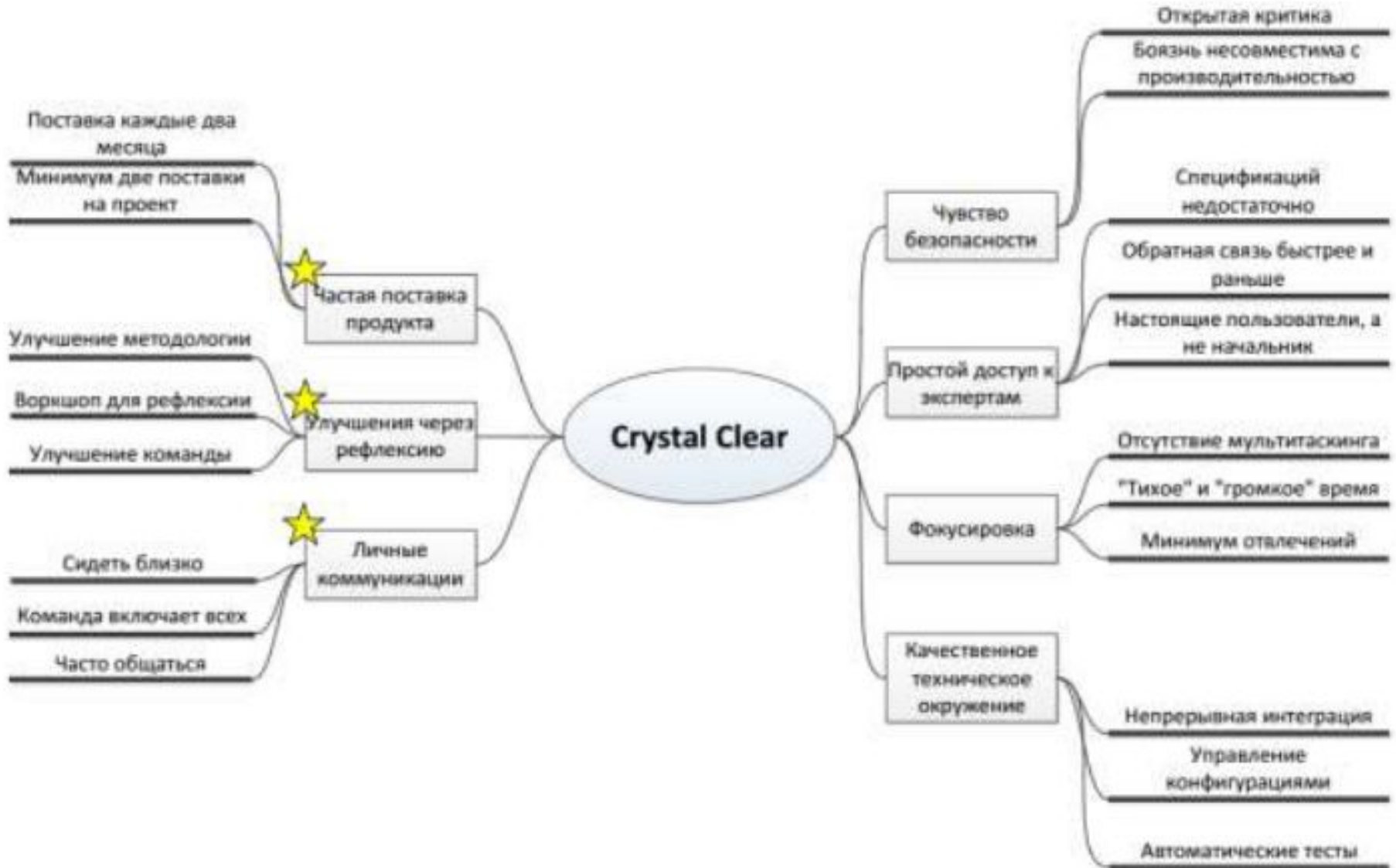
При использовании XP тщательное предварительное проектирование ПО заменяется, с одной стороны, постоянным присутствием в команде заказчика, готового ответить на любой вопрос и оценить любой прототип, а с другой – регулярными переработками кода (так называемый рефакторинг).

Основой проектной документации считается тщательно прокомментированный код.

Очень большое внимание в методологии уделяется тестированию. Как правило, для каждого нового метода сначала пишется тест, а потом уже разрабатывается собственно код метода до тех пор, пока тест не начнет выполняться успешно. Эти тесты сохраняются в наборах, которые автоматически выполняются после любого изменения кода.



# Crystal Clear



# Выводы

В лекции выделены основные этапы жизненного цикла программного обеспечения, их особенности и процессы, которые происходят на каждом этапе. Приведены модели жизненного цикла ПО двух типов: каскадная и спиральная.

Рассмотрен инструментарий технологий программирования:

- языки и системы программирования,
- инструментальные среды пользователя.

Наглядно представлены методологии разработки программного обеспечения:

- Прогнозируемые (RUP),
- адаптивные (Scrum, eXtreme Programming, Crystal Clear).

# Вопросы по лекции

1. Что такое жизненный цикл программного обеспечения?
2. Из каких этапов состоит жизненный цикл программного обеспечения?
3. Что представляет собой требование к программному обеспечению?
4. Каким образом можно собрать требования к программному обеспечению?
5. Для чего нужна спецификация требований? Что она из себя представляет?
6. Какие типы моделей жизненного цикла вы знаете?
7. Какие средства создания приложений бывают?
8. Какие виды языков программирования вы знаете?
9. Назовите 5 современных языков программирования.
10. Какие методологии разработки ПО вы знаете?
11. Назовите ключевые отличия прогнозируемых и гибких методологий разработки ПО.

# Список литературы

1. Калентьев А.А. Новые технологии в программировании : учебное пособие / А.А. Калентьев, Д.В. Гарайс, А.Е. Горяинов – Томск : Эль Контент, 2014. – 176 с. – ISBN 978-5-4332-0185-9.- URL: [http://biblioclub.ru/index.php?page=book\\_view\\_red&book\\_id=480503](http://biblioclub.ru/index.php?page=book_view_red&book_id=480503)
2. Трофимов, В.В. Алгоритмизация и программирование : учебник / В.В. Трофимов, Т.А. Павловская. – Москва : Издательство Юрайт, 2019. – 137 с. – ISBN 978-5-534-07834-3. – URL: <https://www.biblio-online.ru/viewer/algoritmizaciya-i-programmirovanie-423824>