

# • Декларативное планирование

# **Современное состояние дискуссии о разработке ПО**

Две позиции, и актуальные проблемы

# Две позиции

## “Defined” Process (CMMI)

- Повторяемость и воспроизводимость процесса, понимаемого как набор активностей.
- Взгляд на процесс как на изготовление артефактов, акцент на процедуры контроля их качества.
- Точное следование «зрелому» процессу повышает качество результата и снижает риски всей разработки.
- Процесс планируется и контролируется метриками.

## “Empiric” Process (Agile)

- Процесс креатива не воспроизводим и не повторяем, активности могут быть разные.
- Разработка - решение проблем пользователя, акцент на получение ранней обратной связи.
- Bullshit. Следование процессу не может давать никаких гарантий адекватности результата.
- Процесс итеративен и контролируется быстрой обратной связью.

# Обе по своему верны

- Иначе бы не было дискуссии.
- **Планирование и координации деятельности сложных проектов является проблемой.**
- Empiric-agile методы не дают ответа.
- Сетевое планирование и практики PMBoK – исходят из позиции “defined process”, и эти методы плохо подходят для разработки ПО.
- Они из альтернативной реальности, в которой в разработке ПО успешно работает waterfall.

# Разработка ПО

- Неполные и/или противоречивые требования. Не всегда ясно, что надо получить в конце. В том числе – неясно пользователю.
- Неизвестные наперед способы решения проблем. Не всегда ясно, как «это» надо делать. Иногда непонятно, можно ли сделать вообще.
- Неизвестные наперед проблемы. А если кажется, что ясно и понятно, то этому ощущению нельзя доверять.

# Разработка ПО – Problem Solving

- В чем мы вообще можем быть уверены? В постановке проблем.
  - Пользователь обычно **осознает свою проблему**, но не всегда знает, как ее решить. Список требований – описание решения его проблем, которое ставит проблемы разработчику.
  - Разработчик часто знает проблемы, которые мешают ему быстро спроектировать систему. Дизайн – подход к решению этих проблем.
  - QA знает проблемы, мешающие выпуску ПО. База дефектов – описание этих проблем.
- **Разработка в большей степени процесс решения проблем, чем изготовления артефактов.**

# Проблема остается

- Заранее неизвестны:
  - структура результата;
  - часто – технология его «изготовления», ее еще надо придумать;
  - список активностей по его «изготовлению».
- Характер активностей и структура зависят от того, каким образом будут решены проблемы.
- Разработка – сплошные проблемы, и активности со структурой меняются по ходу разработки.
- Планы в терминах активностей трудно составлять, и они быстро перестают быть актуальными.
- **Потребность в планировании и координации деятельности на сложных проектах никуда не делась.**

# Auftragstaktik

## - ПОДХОД ВОЕННЫХ

«Auftragstaktik - странное немецкое слово, обозначающее специфический подход к военному делу. Его почти невозможно перевести на английский, его крайне трудно объяснить, и ещё труднее — понять»



# Две позиции

## Befehlstaktik

- Выдача детализированных приказов в терминах действий.
- Исполнитель выполняет приказы руководителя, проявляя «разумную инициативу».
- Приказ должен быть выполнен точно, беспрекословно, и в срок.
- Следование приказу является оправданием в случае неудачи.

## Auftragstaktik

- Выдача «миссии», проблемы, которая должна быть решена. Выработка действий и решений – обязанность подчиненных.
- Исполнитель обязан руководствоваться собственным видением ситуации и намерением руководителя.
- Бездействие трактуется хуже, чем действие, оказавшееся впоследствии ошибкой.

# Определение Auftragstaktik

- Auftragstaktik – основной принцип управления в армии. Он основан на взаимном доверии, и требует твердого обязательства каждого солдата выполнить свои обязанности.
- Руководитель доводит до подчиненных свои намерения, устанавливает ясные и достижимые цели, и предоставляет необходимые силы и ресурсы.
- Он укажет детали относительно выполнения только в случае, если разные меры, которые служат той же самой цели, должны быть согласованы, или если этого требуют политические или военные ограничения. Он дает своим подчиненным свободу в выполнении их миссии.

# Комментарии к определению

- Начальник обязан определить цель и набор возможностей и ограничений, в рамках которых подчиненный должен выполнить миссию.
- Руководитель предоставляет необходимые ресурсы, и достижение цели становится ответственностью исполнителя. Его навыки, творческий потенциал, и обязательства будут ключевыми элементами успеха.
- Auftragstaktik не просто техника выдачи приказов, это стиль руководства, который неразрывно связан с определенным типом подчиненного и корпоративной культурой.

# Innere Führung

## «внутреннее руководство»

- Подчиненный - свободная личность.
- Его индивидуальность и чувство собственного достоинства уважаются так же, как и его права и свободы.
- Руководство у него исходит изнутри, а не снаружи. За ним не надо следить, его достаточно направлять.
- Только свободный человек будет действовать по собственной воле, проявляя инициативу и двигаемый чувством ответственности.

# Микроменеджмент

- Недопустимо игнорировать промежуточные уровни управления, непосредственно вмешиваясь в работу исполнителей.
- Руководитель, кто пытается указать все детали, **обречен в них запутаться**. Он потеряет контроль над ситуацией и потерпит неудачу.
- Он **подрывает основы их действий**, и повышает риск тактических ошибок, не владея полной информацией.
- И главное, он **теряет их поддержку, и остается с проблемами один на один**.

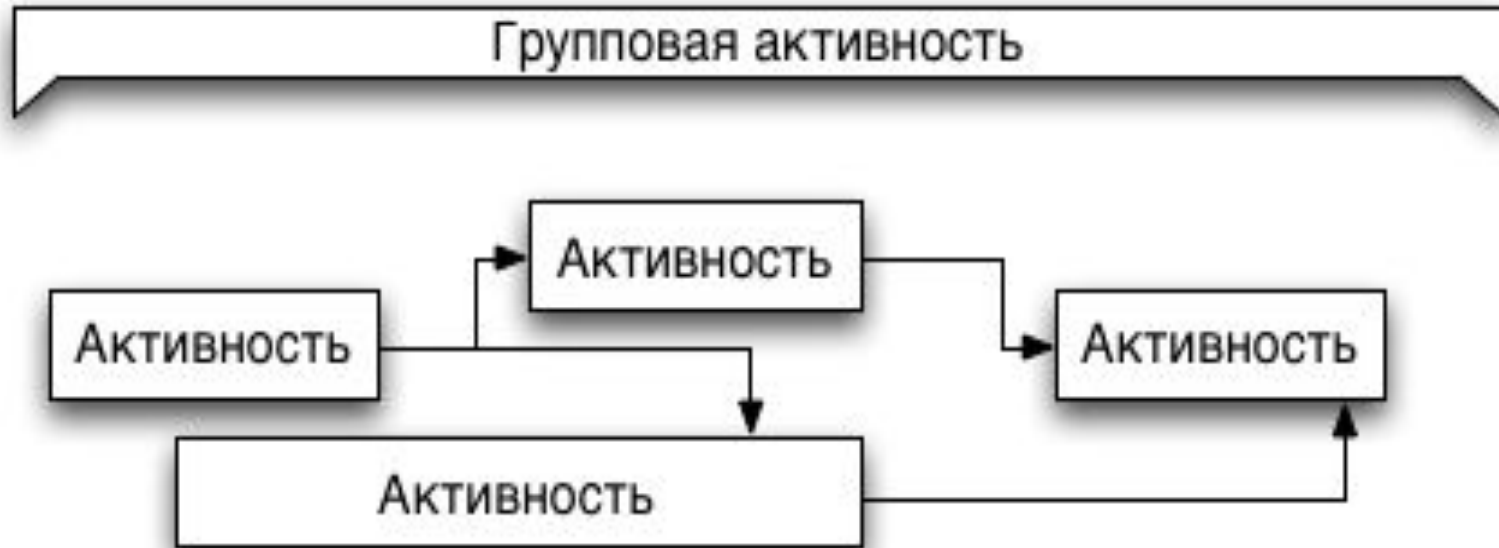
# Микроменеджмент и IT

- С развитием информационных систем скорость передачи информации перестала быть проблемой.
- Это создает ложное впечатление, что у руководителей больше информации для принятия решений, и стимулирует микроменеджмент.
- Однако обработка информации – как была, так и остается проблемой. Возможности человека по ее обработке не изменились.
- **Auftragstaktik - это ключ к успешной обработке информации и принятию решений, и, как следствие, к эффективному управлению.**

# Декларативное планирование

Модель представления плана, дружественная к  
Auftrastaktik.

# Сетевой график = befehlstaktik



- План - как последовательность действий, которые надо выполнить
- Группировка простых действий в более сложные



# АКТИВНОСТИ -> Цели

АКТИВНОСТЬ

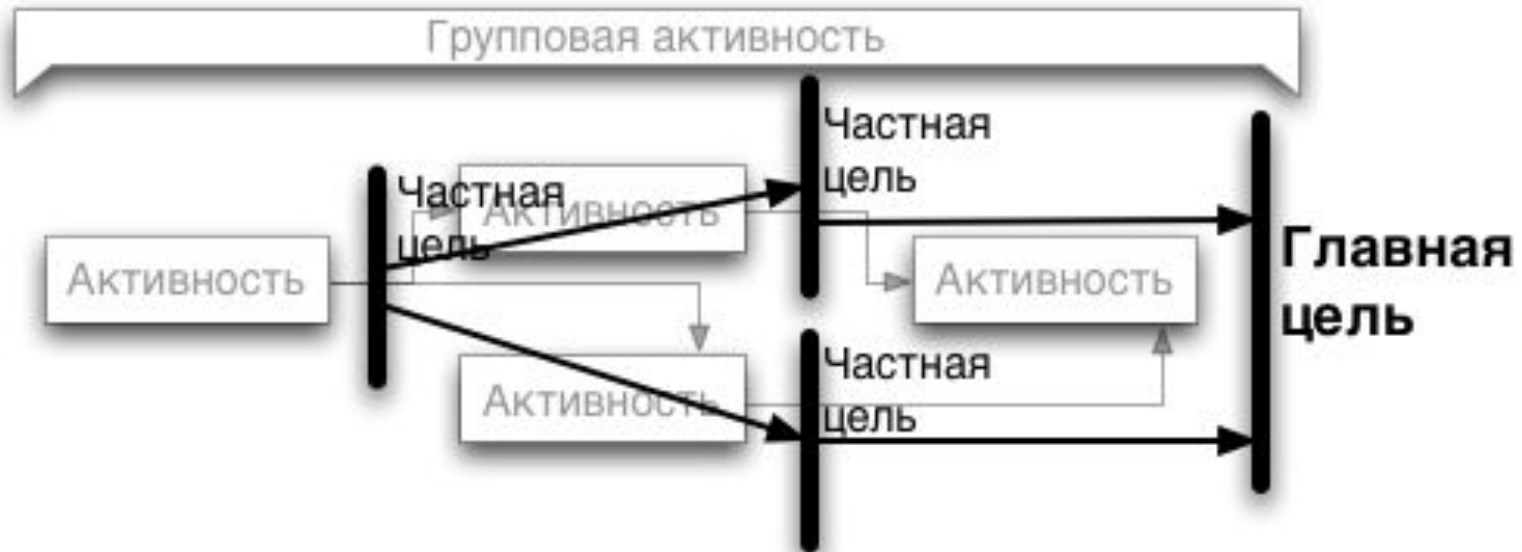
Цель  
Событие  
Результат

## Beheflstaktik

- Активность, протяженная во времени;
- Приказ в терминах действий - что делаем, чем занимаемся.
- 4 типа зависимостей (ОН, ОО, НН, НО)

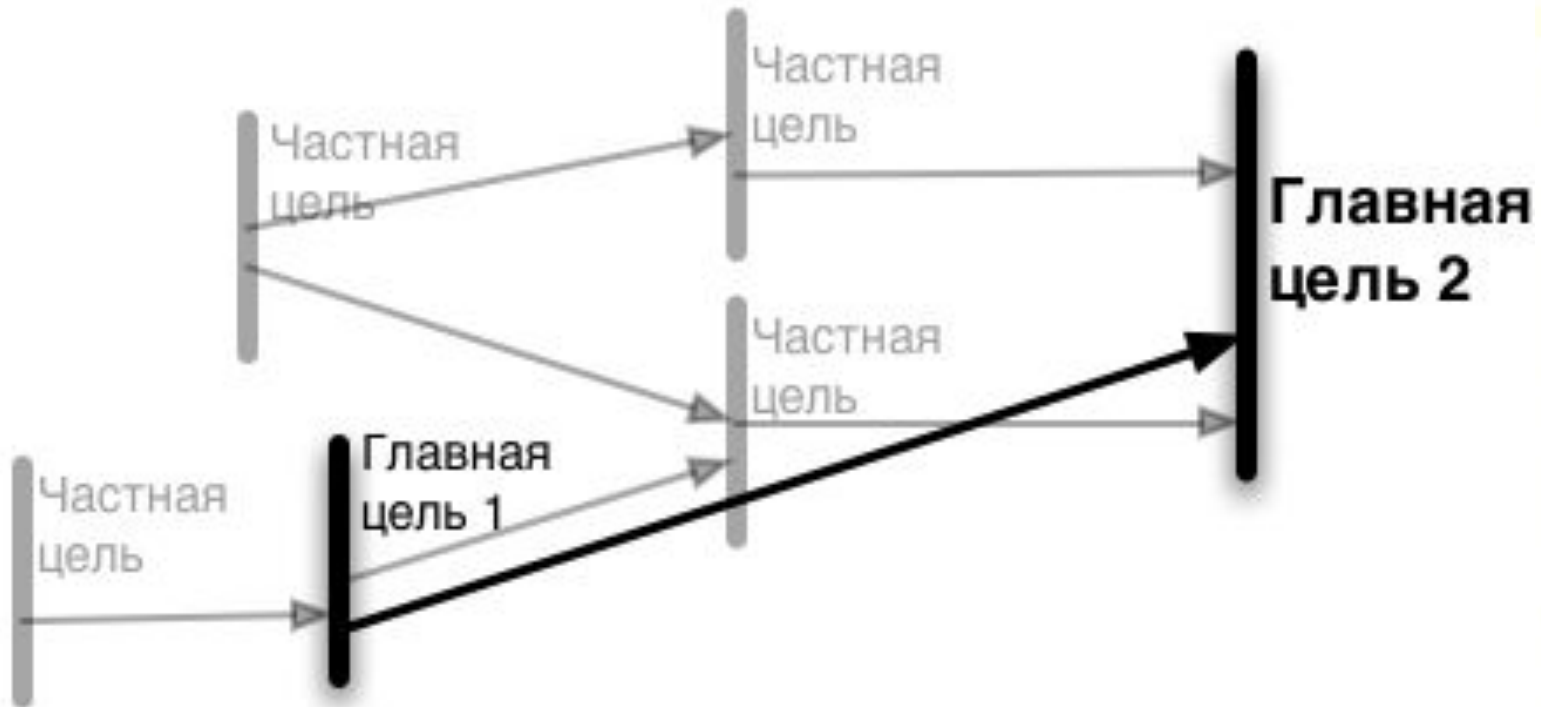
## Auftragstaktik

# «Карта целей» - Auftragstaktik



- План - как группа зависимых целей, которых надо достичь.
- ~~Группировка~~ Отделение главных целей от промежуточных

# Переход к общему плану



- Частные цели могут быть выкинуты из общего плана при сохранении семантики.

# Составление плана «от цели»



- Обратное не всегда верно;
- Поиск верных обратных утверждений – это поиск промежуточных целей, ведущих к главной;
- Выбор подхода к крупной проблеме, подразбивая ее на мелкие – и есть процесс составления плана.

# Свойства

- Фокус на критериях проверки - контроле качества, от “defined process”.
- Фокус на решение проблем – от “empiric process” и auftragstaktik.
- План не ограничивает свободу действий подчиненных. Auftragstaktik!
- План допускает логические проверки.
- Теперь при планировании надо думать. 😊

# **Основные паттерны планирования**

Да, в планах тоже встречаются паттерны.  
Это процессы.

# Цикл разработки RUP

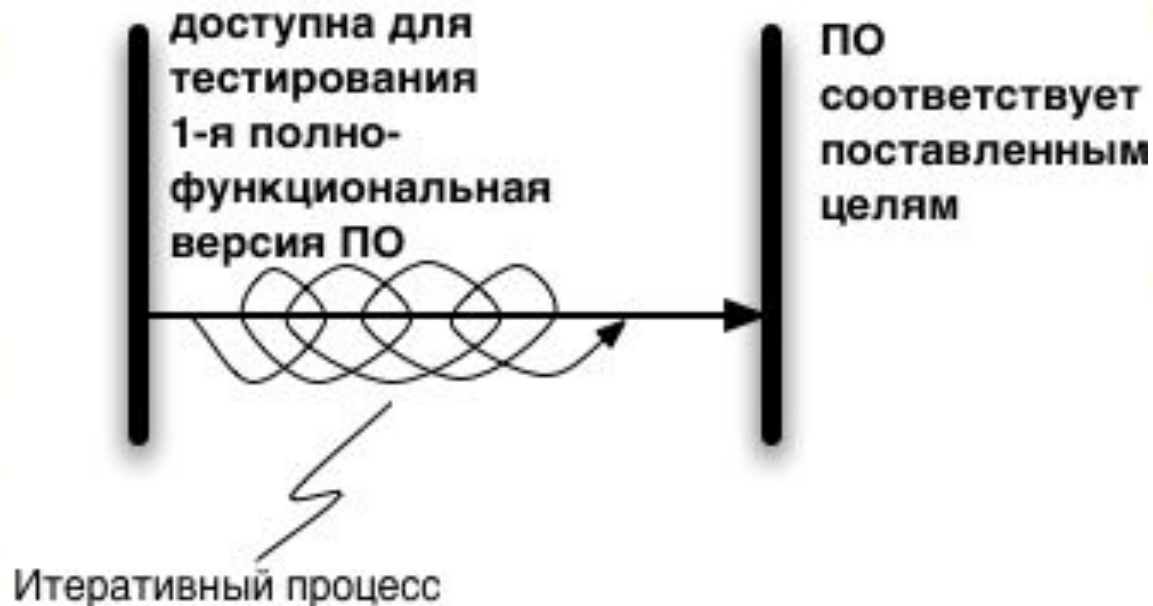


# Процесс решения проблем



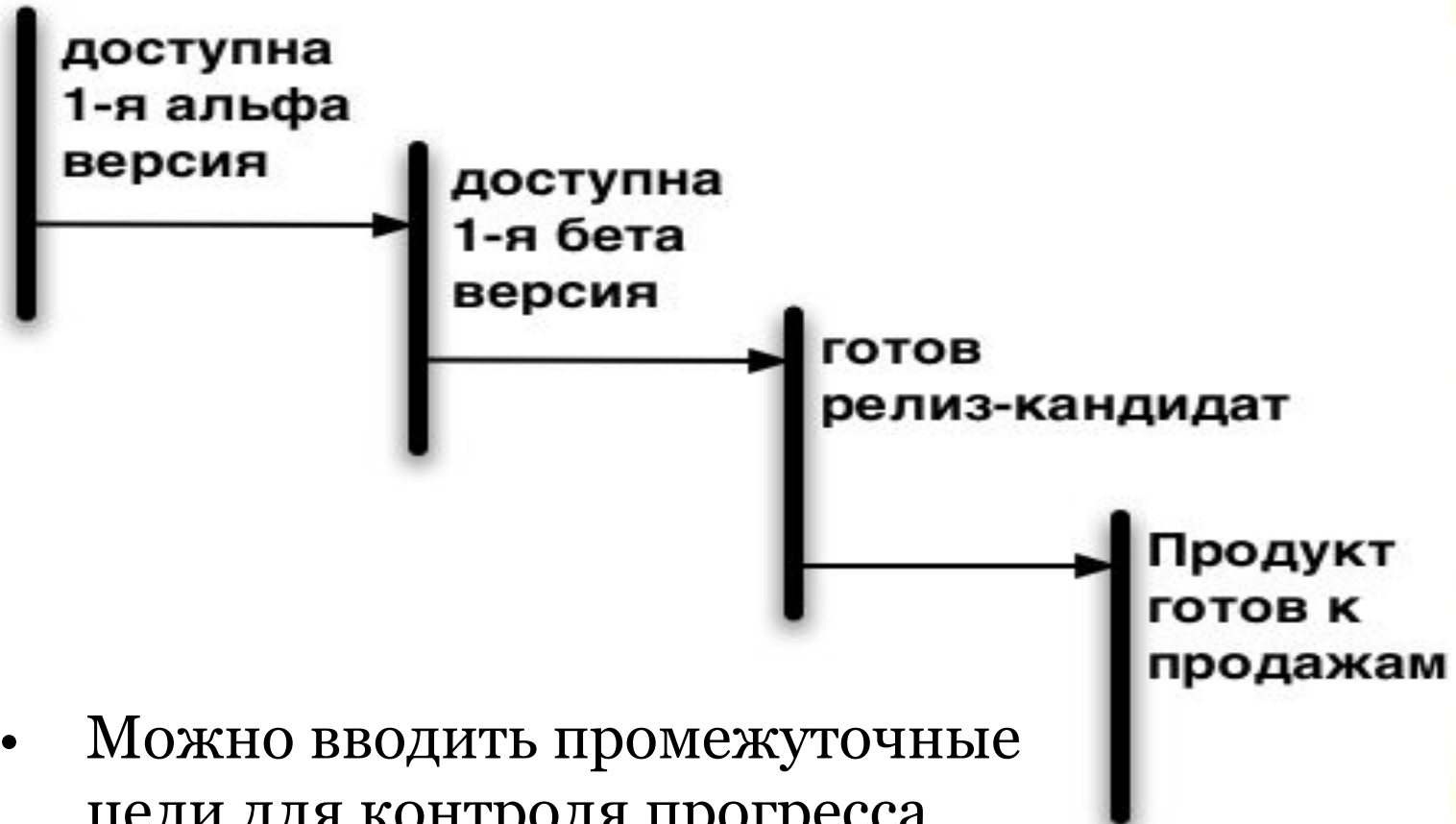


# Итеративные активности



- Итеративные активности полезно «брать в скобки» парой событий «начало-окончание», с четкими критериями достижения.

# Прогресс итеративной активности



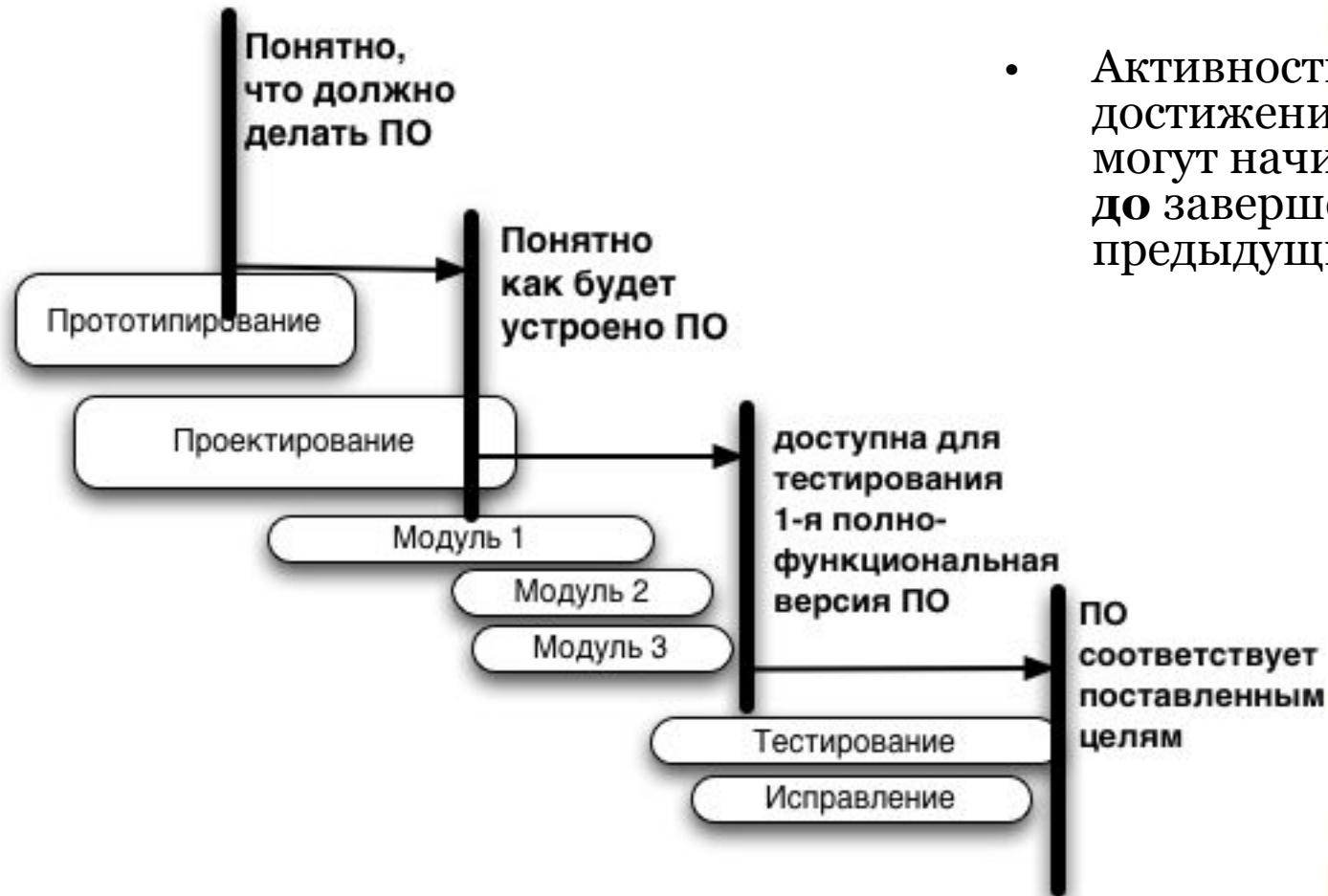
- Можно вводить промежуточные цели для контроля прогресса, если есть четкие критерии их достижения.

# Параллельные процессы



- Один и тот же критерий достижения цели, примененный к разным объектам, может задавать группу независимых целей, достигающихся параллельно.
- Таким образом возможно описание общих процессов, являющихся шаблонами для планов.

# Конвейерные процессы



- Активности по достижению целей могут начинаться **до** завершения предыдущих целей

# Планирование разработки ПО

И как в конце концов составлять план?

# Это все хорошо, но...

- ...как получить не общий абстрактный «процесс», а план для конкретного проекта?
- ...как я могу составить план, если я не знаю, как будет устроена моя программа?
- ...как сделать план, который не разойдется с реальностью в первые же недели?

# ~~Классы и модули?~~

## Функции и тесты!

- Целью является работоспособность **функций** ПО, и эффективное решение проблем пользователя.
- Критерием реализации функции и достижение цели является **прохождение теста (процедуры контроля)** и проверка решения проблемы.
- План – **в каком порядке достигать целей, и как проверять их достижение.**

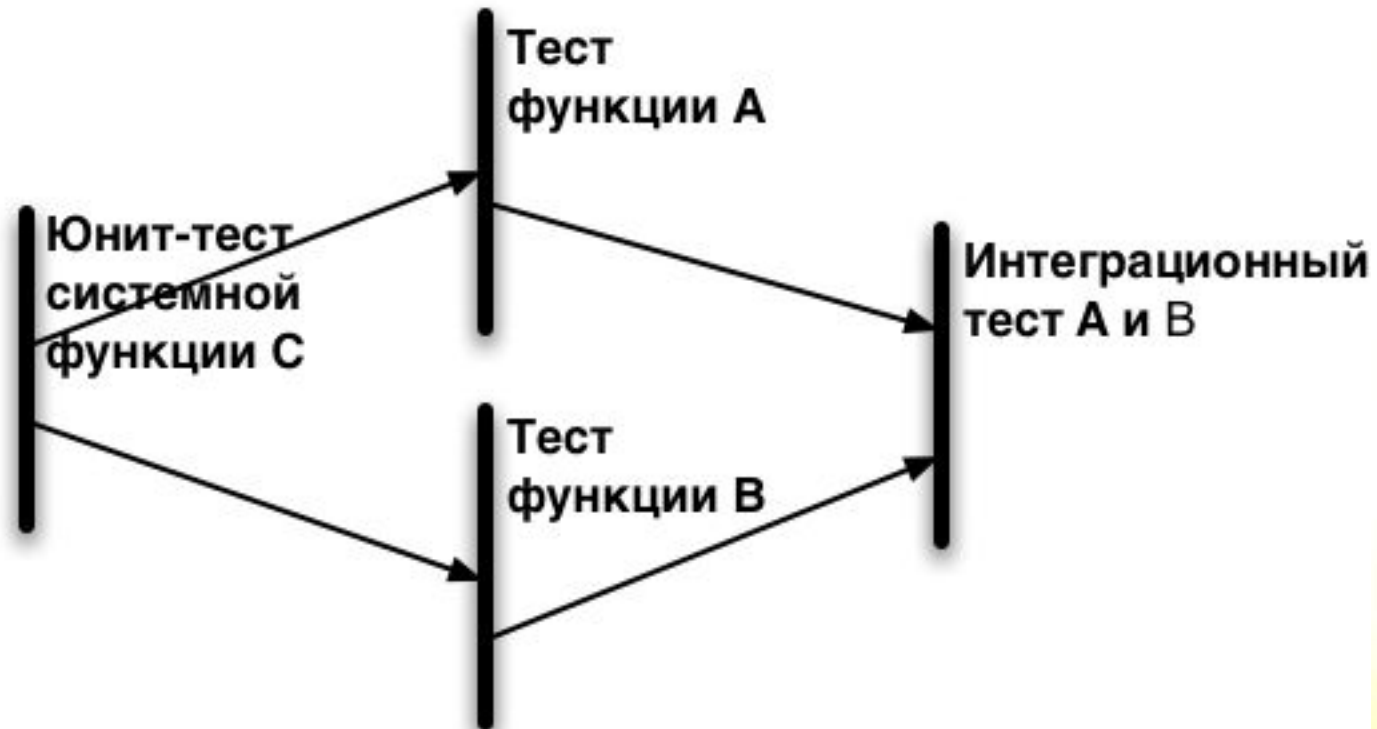
# Функциональный критерий связности

- Цели по реализации функций А и В **несвязны**, если при интеграции реализованных функций не может проявиться ошибок.
- Количество возможных ошибок при интеграции характеризует связность.
- Слабосвязаенные цели могут достигаются впараллель. Сильносвязные – последовательно.



# Общие и «системные» функции

- Пользовательские функции А и В полагаются на общую «системную» функцию С.



# Общие и «системные» функции

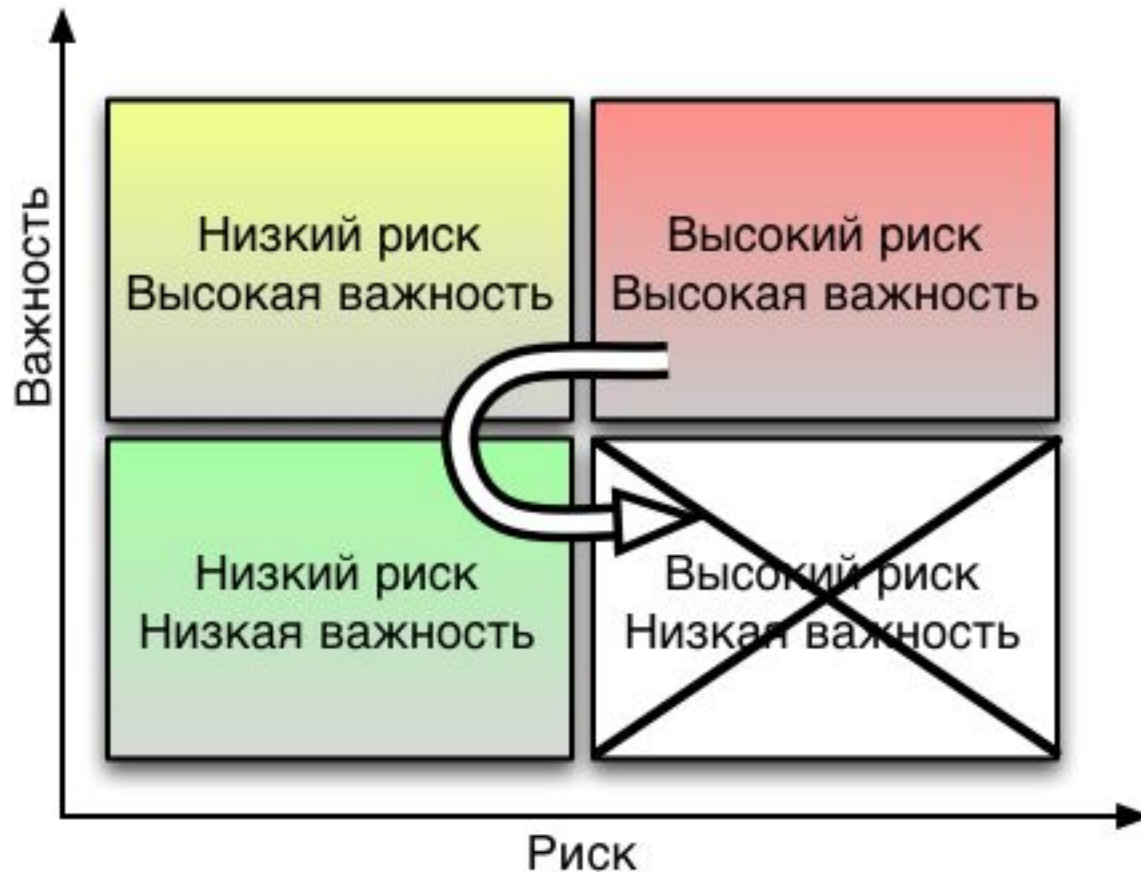
- Пользовательские функции А и В полагаются на общую «системную» функцию С.



- Реализация «системной» функции разделена на два этапа, и делается в рамках работ по пользовательским функциям.

# Выбор порядка реализации функций

- Квадрат риск-приоритет



**Спасибо за внимание!**

Владислав Балин, НТЦ «Модуль»  
gaperton@gmail.com