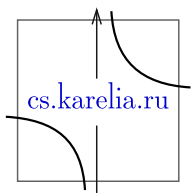
Дисциплина “Руководство процессом разработки программного обеспечения”, осенний семестр 2023 года.

Петрозаводский государственный университет
Институт Математики и информационных технологий
Кафедра Информатики и математического обеспечения

Руководство программным проектом.

Лектор:

к.т.н., доцент Богоявленский Ю. А.
ybgv@cs.karelia.ru



Руководство, общие сведение. Содержание

- Основные понятия руководства проектом
  - Четыре «П» разработки
  - Управление проектом
  - Проблемы управления программными проектами
- Характерные точки руководства проектом
  - Начало проекта
  - Измерения, меры и метрики
  - Процесс оценки
  - Анализ риска
  - Планирование
  - Трассировка и контроль

О планировании. Содержание

- План проекта. Основные характеристики.
- Структура плана: Основные разделы.
- Структура графика работ: Составление графика.
- Структура графика работ: Сбор требований.
- Структура графика работ: Анализ требований.
- Структура графика работ: Сетевая диаграмма работ проекта.
- Структура графика работ: Вычисление границ времени выполнения задачи.
- Структура графика работ: Рекомендации по распределению затрат проекта.

Управление риском. Содержание

- Определение риска.
- Действия по управлению рисками.
- Идентификация риска.
- Анализ риска.
- Ранжирование риска.
- Планирование управления риском.
- Разрешение и наблюдение риска.

Управление персоналом. Содержание

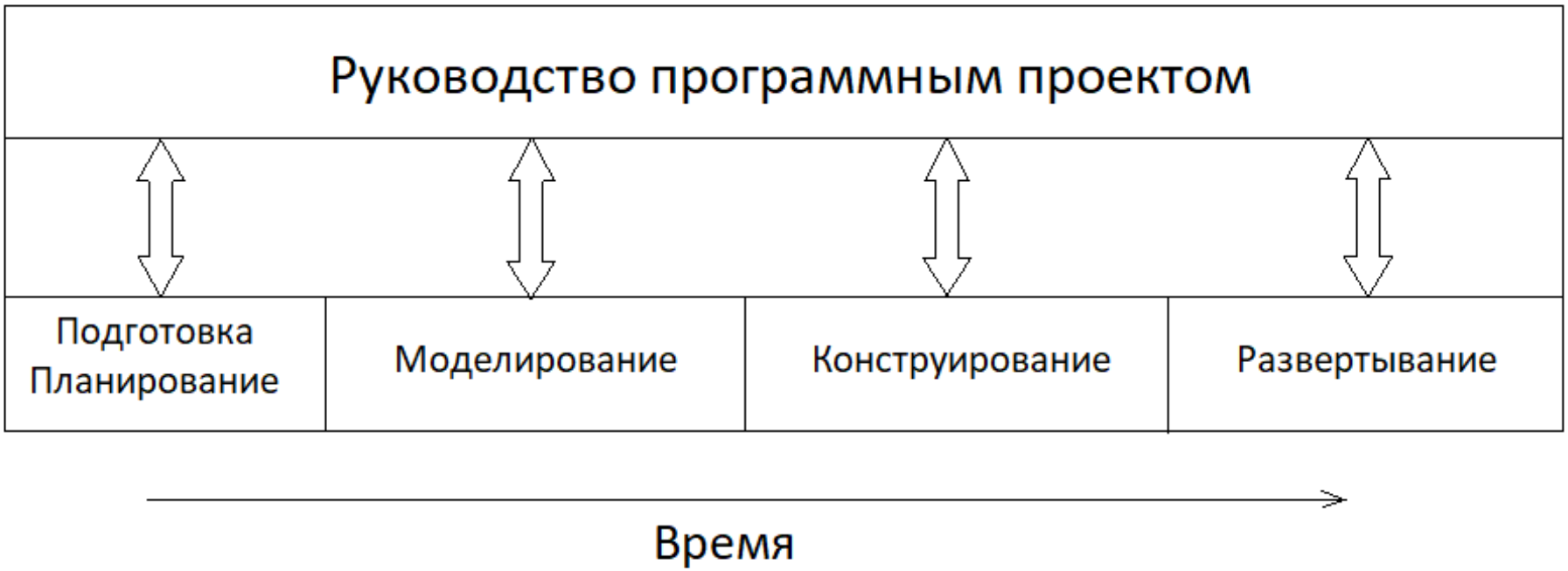
- Управление персоналом
  - Кристаллизация команды
- Подбор членов команды
  - Выбор лидера команды
  - Программирование без персонализации
- Взаимодействия в команде
- Состав группы

Содержание

- Управление документацией:
  - Стандарты и полнота документации.
  - Организации, публикующие общественнозначимые стандарты.
  - Согласованность документации.
- Управление конфигурацией:
  - Определения.
  - Базовые элементы.
  - Задачи управления конфигурацией.
  - Идентификация объектов в конфигурации ПО(базисный и составной объекты.
  - Контроль версий.
  - Контроль изменений.

Основные понятия руководства проектом

Руководство программным проектом — защитная деятельность программной инженерии, пронизывающей все виды основной деятельности.



Основные понятия руководства проектом

Руководство программным проектом является защитной деятельностью программной инженерии, пронизывающей все виды основной деятельности — подготовку, планирование, моделирование, конструирование и развертывание.

Руководство применяется ко всем четырем «П» разработки:

- персоналу (тем, кто это делает);
- процессу (порядку, в котором это делается);
- проекту (среде, в которой это делается);
- продукту (итогу всех дел).

Основные понятия руководства проектом

Цель любого программного *проекта* состоит в производстве определенного программного *продукта*.

«Продукт» включает в себя элементы, называеме *артефактами*:

- программный код;
- документация;
- отчеты по промежуточным итогам;
- результаты проверок;
- оценки качества;
- и т.д.

То, как в рамках проекта создается продукт, представляет собой *процесс*.

Поскольку критичным для успеха дела является взаимодействие членов команды, в рассмотрение включается *персонал*.

Основные понятия руководства проектом

Персонал должен быть организован как эффективная команда, в которой обеспечено эффективное взаимодействие и которая мотивирована на выполнение работы с высоким качеством.

Требования к продукту должны быть переданы (без искажений) от заказчика к разработчикам, правильно разбиты на части и распределены для обработки между инженерами команды.

Процесс должен быть адаптирован как к персоналу, так и к продукту.

Должна быть правильно сконфигурирована среда для выполнения процесса, разумно выбран такой набор рабочих задач, который гарантирует оптимальную загрузку персонала.

Организация проекта должна быть нацелена на успешную работу команды.

Основные понятия руководства проектом: Управление проектом

- *Руководство проектом* заключается в управлении производством продукта в рамках отведенных средств и времени.
- Для управления проектом необходимы не только технические и организационные навыки, но еще *искусство управления людьми*.

Основные понятия руководства проектом

Для проведения успешного проекта нужно понять:

- объем предстоящих работ;
- возможный риск;
- требуемые ресурсы;
- предстоящие задачи;
- прокладываемые вехи;
- необходимые усилия (стоимость);
- план работ.

Руководство программным проектом обеспечивает такое понимание.

Оно начинается перед технической работой, продолжается по мере развития ПО от идеи к реальности и достигает наивысшего уровня к концу работ.

Проблемы управления программными проектами

Проблемы управления программными проектами начали проявляться уже в 60-70-х годах 20-го столетия. Именно в это время заговорили о провалах многих программных проектов. Причем подавляющее большинство провалов связывалось с крупными недостатками в руководстве разработкой.

- Применяемые методики заимствовали опыт управления проектами из других инженерных областей, однако *разработка ПО существенно отличается от разработки физических продуктов*.
- Большие программные проекты чаще всего оригинальны, то есть сильно отличаются от прошлых проектов и являются *инновационными*.
- *Отличительной особенностью* многих программных проектов является предрасположенность к нарушению ограничений по бюджету и времени.

Характерные точки: Начало проекта

Перед планированием проекта следует:

- установить цели и проблемную (предметную) область проекта;
- обсудить альтернативные решения;
- выявить технические и управленческие ограничения.

Без этой информации нельзя обоснованно оценить стоимость, реально распределить задачи проекта, составить такой план руководства проектом, который обеспечивает всестороннее отображение состояния дел.

Характерные точки: Измерения, меры и метрики

Измерения помогают понять как процесс разработки продукта, так и сам продукт.

Измерения процесса производятся в целях его улучшения, измерения продукта — для повышения его качества.

Путем непосредственных измерений могут определяться только опорные свойства объекта.

Типичные вопросы в этой области:

- Как выбрать наиболее подходящие метрики для процессов и продуктов?
- Как использовать полученные данные?
- Корректно ли использовать измерения для сравнения людей, процессов, продуктов?

Характерные точки: Процесс оценки

При планировании программного проекта надо оценить:

- людские ресурсы (в человеко-месяцах);
- продолжительность (в календарных датах);
- стоимость (в тыс. \$).

Характерные точки: Анализ риска

На этой стадии исследуется область неопределенности, имеющаяся в наличии перед созданием программного продукта, и анализируется ее влияние на проект.

Типичные вопросы на данной стадии:

- Действительно ли поняты пожелания заказчика?
- Можно ли реализовать требуемые функции в рамках ограничений проекта?
- Нет ли скрытых от внимания трудных технических проблем?
- Не станут ли изменения, проявившиеся в ходе проекта, причиной недопустимого отставания по срокам?

Характерные точки: Планирование

Вопросы при планировании:

- Предусмотрена ли в плане возможность усовершенствования?
- Не допускает ли план работу в режиме «ошпаренной кошки»?
- Имеется ли резерв времени на непредвиденный случай?
- Есть ли набор хорошо расставленных вех — контрольных рубежей для измерения прогресса?

Работы при планировании:

- Определяется набор задач проекта.
- Устанавливаются связи между задачами.
- Оценивается сложность каждой задачи.
- Определяются людские и другие ресурсы.
- Создается сетевой график задач, проводится его временная разметка.

Характерные точки: Трассировка и контроль

Каждая задача, помеченная в плане, отслеживается руководителем проекта.

При отставании в решении задачи применяются утилиты повторного планирования.

С помощью утилит определяется влияние этого отставания на промежуточную веху и общее время разработки.

В результате повторного планирования:

- могут быть перераспределены ресурсы;
- могут быть реорганизованы задачи;
- могут быть пересмотрены выходные обязательства.

План проекта. Основные характеристики

При работе над программным проектом важным видом деятельности является планирование проекта.

- План помогает предвидеть возможные проблемы разработки ПО и ввести защитные меры для их предупреждения и решения.
- План создается на начальном этапе проекта и рассматривается менеджерами и инженерами-разработчиками как руководящий документ, выполнение которого должно привести к успешному завершению проекта.
- Важными факторами, которые должны учитываться при разработке плана, являются контрактные обязательства фирмы, требования заказчика, бюджетные и временные ограничения.
- План управления проектом должен ясно показать ресурсы, необходимые для воплощения проекта, разделение работ на этапы и временной график выполнения этих этапов.

Последовтельность действий

Рис. 2.2. Последовательность действий при планировании

Структура плана: Основные разделы

План управления проектом должен содержать следующие разделы:

1. Введение;
2. Организация проекта;
3. Анализ рисков;
4. Технический процесс;
5. Распределение работ, график и бюджет.

По мере выполнения проекта план должен регулярно пересматриваться. Одни разделы плана (например, график работ) меняются часто, другие более стабильны.

Структура плана: Введение

1. Обзор проекта. Должен определять проект, но не пытаться охватить все требования к нему. Сами требования приводятся в *Спецификации требований к программному обеспечению*.
2. Результирующие артефакты проекта. Список всех документов, исходных файлов и конечных программных продуктов, которые должны быть созданы.
3. Развитие плана. Направления ожидаемого расширения и изменения.
4. Ссылочные материалы.
5. Определения и аббревиатуры.

Структура плана: Организация проекта

1. Модель процесса. Ссылаются на тип процесса разработки, который будет использован (например, водопадный, спиральный, инкрементальный).
2. Организационная структура. Описывается внутренняя организация команды.
3. Организационные рамки и взаимосвязи. Пути возможного взаимодействия между организациями, что зависит от заинтересованных в проекте сторон. Например, здесь определяется, каким образом будет осуществляться взаимодействие между отделом разработки и маркетинговым, будут ли это регулярные встречи или переписка по электронной почте и т. д.
4. Ответственность за проект. Определяет границы ответственности.

Структура плана: Анализ рисков

1. Цели и приоритеты. Провозглашается рабочая философия проекта.
2. Допущения, зависимости и ограничения.
3. Управление рисками.
4. Механизмы мониторинга и контроля. Определяют, кто будет управлять, контролировать и (или) осуществлять проверку проекта, а также предписывают, как и когда это должно быть сделано.
5. План расстановки кадров.

Структура плана: Технический процесс

1. Методы, инструменты и технологии. Накладываются ограничения на языки и используемые инструменты. Может содержать информацию о повторно используемых требованиях и использовании таких техник, как образцы проектирования.
2. Документация программного обеспечения.
3. Функции сопровождения проекта. Описаны действия для поддержания процесса разработки, такие как управление конфигурацией и обеспечение качества. Если же функция поддержки представлена в различных документах, то в этом пункте будут ссылки на эти документы. В противном случае здесь полностью специфицируются функции поддержки.

Структура плана: Распределение работ, график и бюджет

1. Распределение работ. Поскольку программная архитектура еще не утверждена, первая версия этого пункта будет поверхностной. Детали появляются в последующих версиях плана.
2. Зависимости.
3. Потребности в ресурсах. Оцениваются трудозатраты, аппаратное и программное обеспечение, необходимые для сборки и технической поддержки продукта. Этот пункт уточняется и детализируется с каждой итерацией.
4. Выделение бюджета и ресурсов. Распределяются ресурсы между различными частями проекта в течение всего его жизненного цикла.
5. План-график. Содержит расписание, определяющее, как и когда должны быть выполнены различные этапы процесса.

Структура графика работ. Составление графика

При составлении графика менеджер проекта оценивает длительность проекта, определяет ресурсы, необходимые для реализации рабочих задач, и разворачивает последовательность задач во времени. Как правило, это действие выполняется с помощью специализированной утилиты планирования проекта.

Планирующие утилиты позволяют:

- определить критический путь (цепочку задач, задающих длительность всего проекта);
- определить длительность критического пути;
- установить для каждой задачи наиболее вероятную временную оценку (по прикладной статистической модели);
- вычислить границы, определяющие временное окно для отдельной задачи.

Структура графика работ: Сбор требований

Сбор требований проводится с целью:

1. выяснения потребностей заказчика;
2. оценки выполнимости программной системы;
3. выполнения экономического и технического анализа;
4. определения стоимости и ограничений планирования;
5. создания системной спецификации.

Структура графика работ: Анализ требований

Анализ требований дает возможность:

1. уточнить функции и характеристики программного продукта;
2. обозначить интерфейс продукта с другими системными элементами;
3. определить проектные ограничения программного продукта;
4. построить модели: процесса, данных, режимов функционирования продукта;
5. создать такие формы представления информации и функций системы, которые можно использовать в ходе проектирования.

Структура графика работ: Сетевая диаграмма

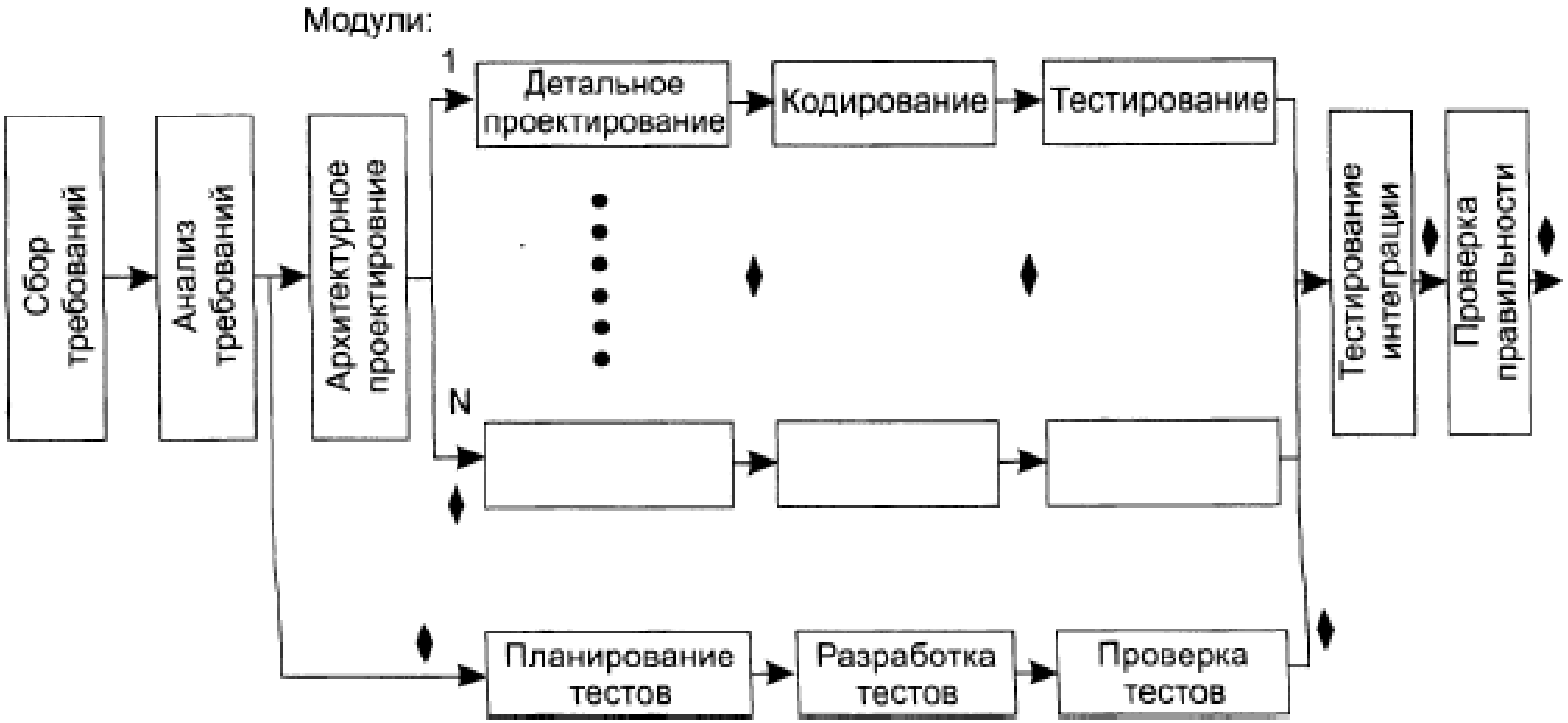


Рис. 2.3. Типовая сетевая диаграмма работ проекта

Структура графика работ: Вычисление границ времени выполнения задачи

Обычно при вычислении границ времени выполнения задачи используются следующие оценки:

1. Раннее время начала решения задачи;
2. Позднее время начала решения задачи;
3. Раннее время конца решения задачи;
4. Позднее время конца решения задачи;
5. Общий резерв — количество избытков и потерь планирования задач во времени, не приводящих к увеличению длительности критического пути.

Структура графика работа. Рекомендации по распределению затрат

Рекомендуемое правило распределение затрат проекта:

- на анализ и проектирование приходится 40% затрат (из них на планирование и сбор требований — 5%);
- на кодирование — 20%;
- на тестирование и отладку — 40%.

Данное правило базируется на следующих фактах:

- Работы, связанные со сбором и формализацией требований, определением и детализацией архитектуры ПО, наиболее трудоемки за счет существенной неопределенности.
- Кодирование (программирование) хорошо проработанных проектных решений имеет меньшую сложность.
- Трудоемкость тестирования и отладки очень велика. Это обусловлено доминирующим стилем человеческой деятельности: «методом проб и ошибок».

Определение риска

Риск- это возможность опасности, неудача. Влияние риска вычисляют по выражению:

$$\mathrm{RE} = \mathrm{P(UO)} \times \mathrm{L(UO)},$$

где

RE — показатель риска (Risk Exposure — подверженность риску);

P(UO) — вероятность неудовлетворительного результата (Unsatisfactory Outcome);

L(UO) — потеря при неудовлетворительном результате.

Действия по управлению рисками

- Этап оценивания риска:
  - Идентификация риска;
  - Анализ риска;
  - Ранжирование риска.
- Этап контроля риска:
  - Планирование управления риском;
  - Разрешение риска;
  - Наблюдение риска.

Идентификация риска

В результате идентификации формируется список элементов риска, а использование проверочных списков помогает выявить возможный риск.

- Дефицит персонала.
- Нереальные расписание и бюджет.
- Разработка неправильных функций и характеристик.
- Разработка неправильного пользовательского интерфейса.
- Интенсивный поток изменения требований.
- Дефицит поставляемых компонентов.
- Дефицит производительности при работе в реальном времени.
- Деформирование научных возможностей.

Идентификация риска

Выделяют три категории источников риска: проектный риск, технический риск, коммерческий риск.

- Источниками проектного риска являются:
  - Выбор бюджета, плана, человеческих ресурсов программного проекта;
  - Формирование требований к программному продукту;
  - Сложность, размер и структура программного проекта;
  - Методика взаимодействия с заказчиком.

Идентификация риска

- К источникам технического риска относят:
  - Трудности проектирования, конструирования, формирования интерфейса, тестирования и сопровождения;
  - Неточность спецификаций;
  - Техническая неопределенность или отсталость принятого решения.

Главная причина технического риска — реальная сложность проблем выше предполагаемой сложности.

Идентификация риска

- Источники коммерческого риска включают:
  - Создание продукта, не требующегося на рынке;
  - Создание продукта, опережающего требования рынка (отстающего от них);
  - Потерю финансирования.

После идентификации элементов риска следует количественно оценить их влияние на программный проект, решить вопросы о возможных потерях. Эти вопросы решаются на шаге анализа риска.

Анализ риска

В ходе анализа оценивается вероятность возникновения Pi и величина потери Li для каждого выявленного i-го элемента риска. В результате вычисляется влияние REi i-го элемента риска на проект.

Таблица 1. Оценка влияния элементов риска.

Элемент риска	Вероятность, %	Потери	Влияние риска
1. Критическая программная ошибка	3-5	10	30-50
2. Ошибка потери ключевых данных	3-5	8	24-40
3. Отказоустойчивость недопустимо снижает производительность	4-8	7	28-56
4. Отслеживание опасного условия как безопасного	5	9	45
5. Отслеживание безопасного условия как опасного	5	3	15
6. Аппаратные задержки срывают планирование	6	4	24
7. Ошибки преобразования данных приводят к избыточным вычислениям	8	1	8
8. Слабый интерфейс пользователя снижает эффективность работы	6	5	30
9. Дефицит процессорной памяти	1	7	7
10. СУБД теряет данные	2	2	4

Ранжирование риска

Каждому элементу риска назначается приоритет, пропорциональный его влиянию проект, что позволяет выделить наиболее важный категории элементов риска.

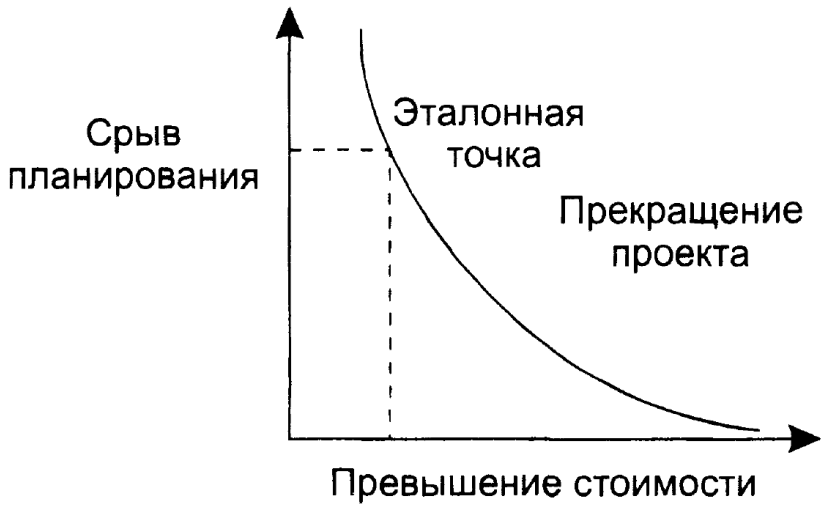
Для больших проектов количество элементов риска может быть очень велико (30-40 элементов), что затрудняет управление риском, поэтому к элементам риска применяют принцип Парето 80/20.

Опыт показывает, что 80% всего проектного риска приходятся на долю 20% от общего количества элементов риска, которые называют существенными элементами, и в дальнейшем учитывается только их влияние.

Планирование управления риском

Цель планирования — сформировать набор функций управления каждым элементом риска. В планировании используют понятие эталонного уровня риска. Обычно выбирают три эталонных уровня риска: превышение стоимости, срыв планирования, упадок производительности. Если комбинация проблем, создающих риск, станет причиной превышения любого из этих уровней, работа будет остановлена.

Рисунок 1. Кривая останова проекта.



Планирование управления риском

Последовательность шагов планирования:

- Исходными данными для планирования является набор четверок: [Ri, Pi, Li, REi], где Ri — i-й элемент риска, Pi — вероятность i-то элемента риска, Li — потеря по i-му элементу риска, REi — влияние i-го элемента риска;
- Определяются эталонные уровни риска в проекте;
- Разрабатываются зависимости между каждой четверкой [Ri, Pi, Li, REi] и каждым эталонным уровнем;
- Формируется набор эталонных точек, образующих сферу останова;
- Для каждого элемента риска разрабатывается план управления;
- План управления каждым элементом риска интегрируется в общий план программного проекта.

Разрешение и наблюдение риска

Основанием для разрешения и наблюдения является план управления риском. Работы по разрешению и наблюдению производятся с начала и до конца процесса разработки.

Разрешение риска состоит в плановом применении действий по уменьшению риска.

Наблюдение риска гарантирует:

- Цикличность процесса слежения за риском;
- Вызов необходимых корректирующих воздействий.

Разрешение и наблюдение риска

Для управления риском используется эффективная методика — «Отслеживание 10 верхних элементов риска». Эта методика концентрирует внимание на факторах повышенного риска, экономит много времени, минимизирует «сюрпризы» разработки. Шаги методики «Отслеживания 10 верхних элементов риска»:

- Выполняется выделение и ранжирование наиболее существенных элементов риска в проекте;
- Производится планирование регулярных просмотров (проверок) процесса разработки;
- Каждый просмотр начинается с обсуждения изменений в 10 верхних элементах риска;
- Внимание участников просмотра концентрируется на любых проблемах в разрешении элементов риска.

Управление персоналом

- Самый ценный ресурс программного проекта - люди
- Предполагается комбинация двух стилей: работа в команде и лидерство. Организация команды, обеспечивающей эффективную работу, является весьма сложной задачей
- Хорошая команда должна демонстрировать сплав самых разнообразных качеств: профессиональные навыки, опыт, сплоченность, дух товарищества. Структура команды должна стимулировать творческую работу всех и каждого.

Кристаллизация команды

- Команда, прошедшая кристаллизацию, — это группа людей, столь сильно связанных, что целое становится больше суммы составляющих его частей
- Как только начинается кристаллизация команды, вероятность успеха возрастает
- До кристаллизации команды ее участники, возможно, имели различные цели. Однако в процессе кристаллизации каждый поверил в общую цель

Подбор членов команды

При работе с кандидатом менеджер обычно учитывает следующие аспекты:

1. опыт работы во многих аппаратно-программных средах
2. знание языков программирования
3. образование и опыт работы по специальности
4. коммуникабельность
5. способность адаптироваться (имеющийся рабочий стаж)
6. жизненная позиция
7. личностные качества

Выбор лидера команды

- отвечает за техническое руководство или за административное управление (возможно и совмещение этих обязанностей)
- должен быть в курсе повседневной деятельности команды, обеспечивая ее эффективную работу и сотрудничество с руководством проекта
- должен ладить со всеми членами коллектива, сглаживая напряженности и неприятности
- «несет знамя командного духа»

Программирование без персонализации

При программировании без персонализации все рабочие продукты (модели, код, документация) считаются собственностью всей команды.

Преимущества разработки без персонализации:

- упрощение процедур проверки, критики недостатков, повышение их объективности
- поощрение стиля непринужденного обсуждения рабочих заданий, достоинств и недостатков отдельных решений
- активизация дружеских отношений, повышение уровня искренности
- быстрый рост мастерства (благодаря работе бок о бок)
- улучшение качества, совершенствование результатов работы

Взаимодействия в команде

Размер/структура команды

- С ростом числа участников количество связей по взаимодействию растет квадратично. Например, между тремя участниками есть три связи, четыре участника имеют шесть связей, пять человек — десять связей, то есть n человек имеют следующее количество связей (каждый с каждым):

$$(n-1)+(n-2)+\ldots+1=\frac{n(n-1)}{2}$$

- Для больших команд альтернативой является их разделение на группы. Каждая группа отвечает за определенную часть проекта и работает над ней. Для взаимодействия с другими группами в каждой группе выделяется один сотрудник.

Взаимодействия в команде

Иерархия команды

- Сотрудники в команде с горизонтальной организацией (один уровень, все сотрудники равны) легче общаются между собой, чем в командах с многоуровневой организацией и иерархией отношений (начальники/подчиненные). В последних командах взаимодействие происходит между уровнями, в иерархической последовательности.

Рабочее окружение

- Человек всегда предпочитает работать в отдельном помещении, которое он может оформить по-своему вкусу и в котором может сосредоточиться.
- В открытом помещении сотруднику сконцентрироваться труднее, следствием чего становится снижение рабочей активности

Состав группы

- Аналитик — отвечает за развитие и интерпретацию требований заказчика; должен быть экспертом в предметной области, но работать в тесном контакте с остальными сотрудниками.
- Архитектор — впередсмотрящий; отвечает за проектирование и развитие архитектуры продукта, является одним из наиболее квалифицированных специалистов, имеющих опыт принятия стратегических решений; кроме опыта проектирования, архитектор должен уметь программировать, поскольку его решения воплощаются в программном коде.

Состав группы

- Конструктор компонентов — главный создатель компонентов.
- Специалист по применению — программирует компоненты и отвечает за их сборку.
- Специалист по повторному использованию — внедряет в продукт готовые компоненты из сторонних коммерческих библиотек.
- Специалист по интеграции — отвечает за сборку совместимых версий компонентов и проверку правильности их совместной работы, поддерживает выпуск версий продукта.

Состав группы

- Специалист по документации — документирует все реализованные решения, готовит документацию для пользователя.
- Системный программист — отвечает за создание и адаптацию программных утилит, облегчающих разработку в проекте.
- Системный администратор — управляет физическими компьютерными ресурсами в проекте.

Не каждый проект требует исполнения всех этих ролей. В небольших проектах сотрудники могут играть сразу несколько ролей.

Управление документацией

Цель — полнота и согласованность документации.

Полнота подразумевает, что комплект документации должен охватывать весь жизненный цикл ПО.

Согласованность означает, что комплект документов не содержит внутренних противоречий (описание факта $\Rightarrow$ только в одном документе $\Rightarrow$ изменения только в одном месте).

Разработка программного обеспечения поддерживается документацией. Чтобы документация была понятна всем, она должна соответствовать принятым стандартам.

Организации, публикующие общественнозначимые стандарты:

- Государственный комитет Российской Федерации по стандартизации и метрологии ;
- Международная организация по стандартизации(ISO);
- Институт инженеров по электротехнике и радиоэлектронике (IEEE) ;
- Институт программной инженерии (SEI).

Управление конфигурацией

Управление конфигурацией — это координация различных версий и частей документации и программного кода. Управление конфигурацией ПО — защитная деятельность, применяемая на всех этапах жизненного цикла ПО.

- В Управление конфигурацией входят:
  - идентификация изменения;
  - контроль изменения;
  - гарантия правильной реализации изменения;
  - формирование сообщения об изменениях.

- Категории информации в составе конфигурации ПО:
  - компьютерные программы (в виде исполняемых кодов);
  - документы, описывающие программы (как для технического персонала, так и для пользователей). ;
  - структуры данных (как внешние, так и внутренние);

Базовые элементы конфигурации

- Системная спецификация.
- План программного проекта.
- Спецификация требований к ПО. Работающий или бумажный макет.
- Предварительное руководство пользователя.
- Спецификация проектирования.
- Листинги исходных текстов программ.
- План и методика тестирования. Тестовые варианты и полученные результаты.
- Руководства по работе и инсталляции.
- Исполняемый код программ.
- Описание базы данных.
- Руководство пользователя по настройке.
- Документы сопровождения. Отчеты о проблемах ПО. Запросы сопровождения. Отчеты об изменениях.
- Стандарты и методики разработки ПО.

Задачи управления конфигурацией

- Идентификация объектов.
- Контроль версий.
- Контроль изменений.
- Проверка конфигурации и отчетность.

Объекты идентифицируемые в конфигурации ПО.

Идентифицируются два типа объектов: базисные и составные объекты.

Базисный объект —– элемент информации, создаваемый в ходе анализа, проектирования, кодирования или тестирования. Составной объект — коллекция базисных объектов и других составных объектов.

Каждый объект имеет набор характеристик, которые определяют его уникальность: имя, описание, список ресурсов и реализацию.

Контроль версий

Контроль версий объединяет процедуры и средства для управления различными версиями объектов конфигурации, которые создаются в ходе разработки.

- В систему контроля версий обычно входят:
  - база данных проекта;
  - средство управления версиями, сохраняющее все версии объектов конфигурации;
  - устройство генерации, позволяющее собирать все значимые объекты конфигурации и создавать определенную версию ПО.

Контроль изменений

При разработке больших систем неконтролируемые изменения быстро приводят к хаосу. Контроль изменений обеспечивается механизмом, включающим человеческие действия и автоматические средства.

Шаги процесса контроля и проведения изменения ПО

- Формируется донесение об изменении.
- Специалист по контролю принимает решение: отвергнуть или утвердить.
- Для обработки заказа объект выбирается из базы данных.
- Выполняется «выходная» проверка объекта.
- Выполняется изменение объекта.
- Проверяется правильность и качество проведенного изменения.
- Выполняется «входная» проверка объекта, после чего он заносится в БД.
- Обновленный объект включается в следующую реализацию.
- Перестраивается соответствующая версия ПО.
- Проверяются изменения всех элементов конфигурации.
- Изменения включаются в новую версию.
- Выпускается новая версия ПО.

Спасибо за внимание