

## Управление программным проектом. Планирование и риски

Алексей Островский

Физико-технический учебно-научный центр НАН Украины

7 мая 2015 г.

# Управление программным проектом

## Определение

**Управление программным проектом** (англ. *software project management*) — деятельность, направленная на производство качественного программного обеспечения при ограниченных бюджетных и временных ресурсах.

### Цели управления:

- ▶ поставка ПО в заданный срок;
- ▶ соблюдение расходов в рамках бюджета;
- ▶ поставка ПО, удовлетворяющего запросы заказчика;
- ▶ организация продуктивной команды разработки.

### Проблемы управления производством ПО:

- ▶ неосвязаемость продуктов производства;
- ▶ уникальность программных проектов;
- ▶ специфичность процессов производства ПО.

## Составляющие управления

### ▶ Планирование проекта:

- ▶ планирование, оценка и составление планов по разработке ПО;
- ▶ назначение ответственных за выполнение заданий;
- ▶ контроль выполнения заданий.

### ▶ **Отчетность:** составление отчетов о ходе выполнения проекта для заказчика и организации-разработчика.

### ▶ Управление рисками:

- ▶ оценка рисков, влияющих на проект;
- ▶ мониторинг рисков;
- ▶ устранение проблем, связанных с рисками.

### ▶ Управление персоналом:

- ▶ подбор персонала для проекта;
- ▶ организация коммуникации между разработчиками.

### ▶ Создание **проектных планов для заключения контракта** с заказчиком.

# Риски

## Определение

**Риск** — нежелательное событие при выполнении проекта, которое может иметь непредвиденные отрицательные последствия.

### Категории рисков:

- ▶ проектные риски — риски, связанные с расписанием проекта или ресурсами (напр., потеря опытного разработчика);
- ▶ риски, связанные с продуктом — риски, влияющие на качество или производительность ПО (напр., неудовлетворительное поведение приобретенного компонента);
- ▶ бизнес-риски — риски, влияющие на организацию-разработчика (напр., выход на рынок конкурирующего продукта).

## Примеры рисков

Категория	Риск
Проектные	смена персонала; изменение структуры управления; отсутствие необходимого оборудования.
Проектные и производственные	изменение требований; задержки с определением спецификации; недооценка размера и сложности системы.
Производственные	недостаточная эффективность CASE-инструментов.
Бизнес	устаревание базовых технологий; появление конкурирующих продуктов.

# Управление рисками

## Этапы управления рисками:

1. Идентификация рисков.

**Результат:** перечень потенциальных рисков.

2. Анализ рисков.

**Результат:** установка приоритетов для рисков.

3. Планирование рисков.

**Результат:** планы по избеганию и минимизации рисков.

4. Мониторинг рисков.

**Результат:** оценка актуальных рисков на данный момент работы с проектом.

## Идентификация рисков

- ▶ **Технологические риски** (связанные с оборудованием или ПО, используемым в разработке).

### Примеры:

- ▶ недостаточная скорость обработки транзакций СУБД;
- ▶ дефекты в компонентах повторного использования.

- ▶ **Кадровые риски.**

### Примеры:

- ▶ невозможность набора компетентного персонала;
- ▶ отсутствие должных навыков у членов команды разработки;
- ▶ недоступность ключевого персонала (напр., по причине болезни) в ключевые периоды.

- ▶ **Организационные риски** (связанные с организацией разработки).

### Примеры:

- ▶ реструктуризация управления проектом во время разработки;
- ▶ сокращение финансирования проекта.

## Идентификация рисков (продолжение)

- ▶ **Инструментальные риски** (вызванные инструментами разработки).

### Примеры:

- ▶ неэффективность кода, сгенерированного автоматически;
- ▶ невозможность интеграции различных инструментов разработки.

- ▶ **Риски, связанные с требованиями.**

### Примеры:

- ▶ необходимость кардинального изменения архитектуры, связанная с изменениями требований.

- ▶ **Оценочные риски** (связанные с неправильной оценкой ресурсов).

### Примеры:

- ▶ недооценка времени, требуемого на разработку;
- ▶ переоценка скорости исправления дефектов системы;
- ▶ недооценка размера разрабатываемого ПО.



## Анализ рисков

### Вероятность рисковой ситуации:

- ▶ очень низкая ( $< 10\%$ );
- ▶ низкая (10–25 %);
- ▶ средняя (25–50 %);
- ▶ высокая (50–75 %);
- ▶ очень высокая ( $> 75\%$ ).

### Последствия рисковой ситуации:

- ▶ катастрофические (угроза существованию проекта);
- ▶ серьезные (большие задержки);
- ▶ терпимые (задержки в пределах плана);
- ▶ незначительные.

# Планирование рисков

## Стратегии реагирования на риски:

- ▶ **Избегание рисков** — минимизация возникновения рисковой ситуации.

**Риск:** дефективные компоненты.

**Стратегия:** использование проверенных компонентов (напр., использованных в предыдущих проектах).

- ▶ **Минимизация последствий.**

**Риск:** недоступность персонала в ключевой период разработки.

**Стратегия:** реорганизация разработки для усиления взаимодействия между разработчиками и повышения понимания сути работы коллег.

- ▶ **План действий** в чрезвычайной ситуации.

**Риск:** сокращение финансирования проекта.

**Стратегия:** подготовка документа для начальства с описанием важности проекта для организации и нецелесообразности сокращения бюджета.

# Мониторинг рисков

## Определение

**Мониторинг рисков** — проверка базовых предположений о рисках, их вероятности и влияния на процесс разработки.

Типы рисков	Факторы
Технологические	задержки в доставке ПО или оборудования; проблемы работы с ними.
Кадровые	плохие отношения в команде разработки; высокая текучесть кадров.
Организационные	бездействие начальства.
Инструментальные	недостаточное использование инструментов; запрос на более мощные рабочие компьютеры.
Требования	большое число запросов на изменение требований.
Оценочные	отставания в расписании; низкая скорость устранения дефектов.

# Планирование проекта

## Задачи:

- ▶ разбиение процесса разработки на составляющие;
- ▶ назначение ответственных за выполнение заданий;
- ▶ оценка затрат на выполнение проекта;
- ▶ управление рисками;
- ▶ оценка текущей степени выполнения проекта.

## Этапы планирования:

1. предварительный план (составляется до подписания контракта с заказчиком);
2. начальный план (составляется в начале разработки);
3. периодические планы (уточняются по ходу выполнения проекта).

# Разработка через планирование

## Определение

**Разработка через планирование** (англ. *plan-driven development*), **классическая модель разработки** — подход к разработке ПО, основанный на детальном планировании процессов производства.

### Достоинства:

- ▶ раннее выделение организационных проблем (доступность персонала, взаимодействие с другими проектами);
- ▶ ранняя оценка затрат на проект;
- ▶ упрощение взаимодействия с заказчиком и третьими сторонами (напр., для сертификации).

### Недостатки:

- ▶ громоздкость системы документирования проекта;
- ▶ высокие затраты на внесение изменений по ходу проекта;
- ▶ низкая степень адаптации к изменениям в среде выполнения.

# Проектный план

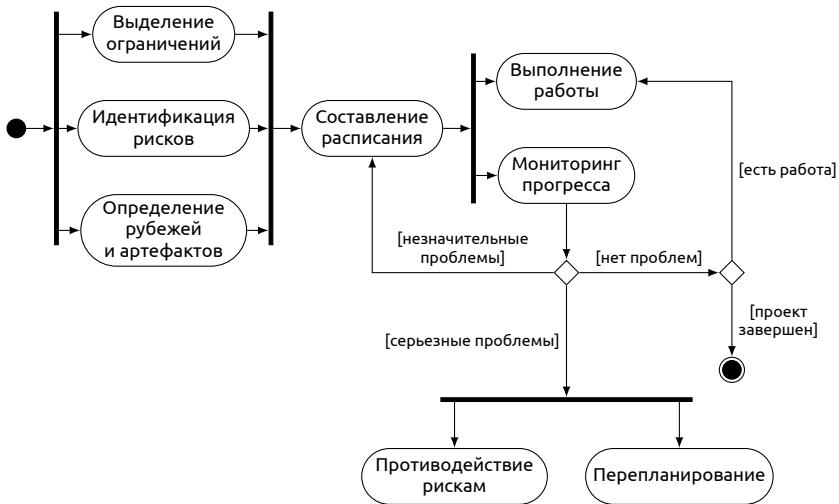
## Составляющие плана:

1. **Введение** — решаемые задачи; ограничения (бюджетные, временные, ...), влияющие на управление проектом.
2. **Организация проекта** — организация команды разработки, роли разработчиков.
3. **Анализ рисков** — риски, вероятность их появления, стратегии реагирования.
4. **Ресурсные требования** на оборудование и ПО, оценка стоимости приобретения.
5. **Разбиение работы** на составляющие процессы, выделение рубежей (англ. *milestone*) и артефактов производства (англ. *deliverable*).
6. **Расписание проекта** — зависимости между процессами, оценка времени выполнения, распределение кадров.
7. Механизмы **мониторинга** и **отчетности**.

## Дополнительные планы

- ▶ **План качества** — описание деятельности по достижению, измерению и контролю качества процессов и артефактов производства.
- ▶ **План валидации** — описание подхода к проверке программной системы, ресурсов и расписания валидации.
- ▶ **План управления конфигурацией** — описание методов управления конфигурацией проекта и используемых утилит (систем управления версиями, утилит построения, ...).
- ▶ **План сопровождения** — прогноз требований, затрат и стоимости сопровождения проекта.
- ▶ **План квалификации кадров** — описание повышения квалификации и опыта разработчиков по ходу выполнения проекта.

## Процесс планирования





# Составление расписания проекта

## Определение

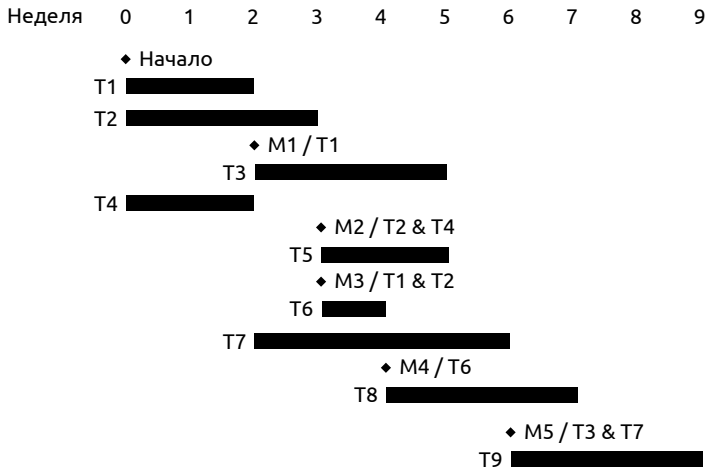
**Процесс** (англ. *activity*) — автономная часть разработки, которая характеризуется:

- ▶ длительностью (1–8 недель);
- ▶ оценкой объема работ (в человеко-днях);
- ▶ граничным сроком завершения;
- ▶ условием завершения (напр., удачное выполнение всех тестов).

## Этапы составления расписания:

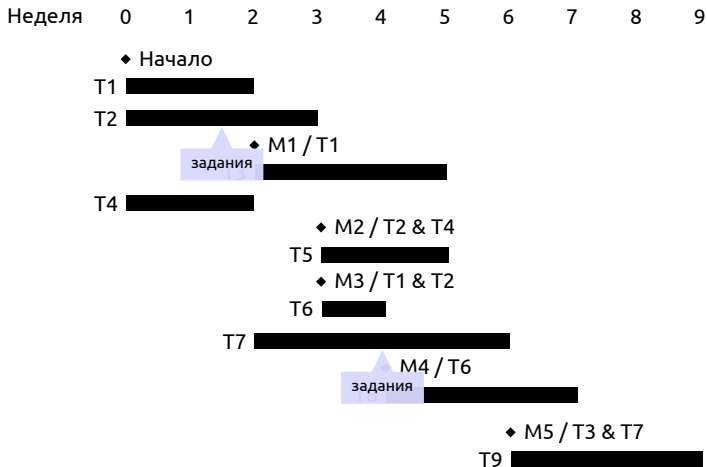
1. выделение процессов на основе требований и предварительной архитектуры;
2. определение взаимодействия и зависимостей между процессами;
3. оценивание ресурсов для выполнения процессов;
4. распределение членов команды разработки по процессам;
5. создание расписания и его визуальных представлений (графиков, диаграмм, ...).

## Представление расписания



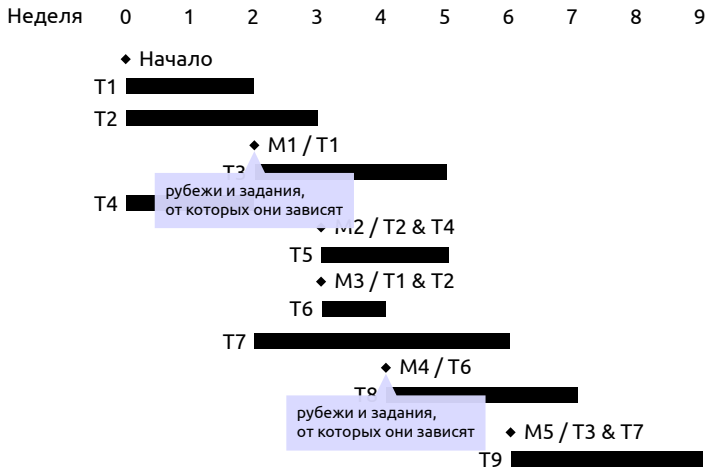
Представление расписания в виде столбчатой диаграммы

## Представление расписания



Представление расписания в виде столбчатой диаграммы

## Представление расписания



Представление расписания в виде столбчатой диаграммы

# Гибкое планирование

## Определение

**Гибкое планирование** (англ. *agile planning*) — составление инкрементальных итеративных планов по производству программного продукта по ходу разработки.

**Базовое предположение:** требования заказчика уточняются и изменяются в ходе разработки ПО.

### Этапы планирования:

1. Планирование выпусков — определение характеристик следующего выпуска программной системы.

**Длительность:** несколько месяцев.

2. Планирование итераций — определение следующей итерации разработки.

**Длительность:** 2–4 недели.

# Планирование в XP

## Этапы планирования в экстремальном программировании (XP):

1. **выделение сценариев использования** (англ. *user story*), по возможности покрывающих всю функциональность системы;
2. **оценка затрат** на имплементацию функций, соответствующих различным сценариям;
3. **оценка скорости реализации** (из предыдущего опыта или путем имплементации нескольких тестовых сценариев);
4. **планирование выпуска:** подбор и конкретизация сценариев для следующего выпуска системы (совместно с заказчиком);
5. **планирование итерации:** подбор сценариев для итерации с учетом скорости реализации, разбиение сценариев на короткие задания (4–16 часов);
6. **оценка прогресса** посередине итерации для удаления «отстающих» сценариев из итерации.

# Оценка затрат

## Проблемы оценки затрат на процессы разработки:

- ▶ большое количество неизвестных параметров на ранних этапах (новые технологии, навыки команды разработки);
- ▶ невозможность проверки корректности оценки.

## Методы оценки:

- ▶ **Экспертные методы** (на основе опыта управления предыдущими проектами) — оценка затрат на производство отдельных артефактов и суммирование получившихся оценок.

**Проблемы:** невозможность оценки при использовании новых технологий.

- ▶ **Алгоритмические методы** — оценка затрат на основе метрик проекта и процессов (напр., размер проекта).

## Алгоритмическое моделирование затрат

### Общая формула оценки затрат:

$$Effort = A \times Size^B \times M.$$

- ▶  $A$  — константа, зависящая от типа разрабатываемого ПО и организации работы;
- ▶  $Size$  — оценка размера ПО или его сложности;
- ▶  $B$  — показатель, зависящий от сложности проекта ( $1 \leq B \leq 1,5$ );
- ▶  $M$  — множитель, зависящий от характеристик проекта (напр., требований к безотказности ПО или навыков команды разработки).

### Проблемы:

- ▶ сложности с оценкой размера  $Size$  на ранних этапах;
- ▶ субъективность факторов, влияющих на  $B$  и  $M$ .



## Пример: модель COSOMO II

### Определение

**COSOMO II** (constructive cost model) — модель для оценки затрат на производство ПО, разработанная Б. Бёмом (Barry Boehm).

### Подмодели:

- ▶ **Модель композиции** (англ. *application composition model*) — оценка затрат на построение системы из КПИ, с помощью сценариев и программирования БД.
- ▶ **Модель раннего проектирования** (англ. *early design model*) — предварительная оценка затрат на разработку на основе описания интерфейсов системы.
- ▶ **Модель повторного использования** (англ. *reuse model*) — оценка затрат на интеграцию КПИ и / или автоматически сгенерированного кода.
- ▶ **Пост-архитектурная модель** (англ. *post-architecture model*) — оценка затрат на основе характеристик проекта.

## Выводы

1. Управление рисками и планирование проекта — две важных составляющих управления разработкой ПО, наряду с управлением качеством и управлением конфигурацией.
2. Управление рисками включает оценку вероятности и последствий рискованных ситуаций, разработку стратегий борьбы с ними, а также мониторинг рисков по ходу выполнения проекта.
3. Планирование проекта позволяет разбить разработку на мелкие процессы, определить расписание их выполнения и ответственных разработчиков.
4. Оценка затрат — одна из ключевых задач планирования разработки ПО. Для ее решения существуют алгоритмические методы, в т. ч. COCOMO (constructive cost model).

# Материалы



Sommerville, Ian

Software Engineering.

Pearson, 2011. — 790 p.



Лавріщева К. М.

Програмна інженерія (підручник).

К., 2008. — 319 с.

Спасибо за внимание!