

## Лекция 6: Планирование проекта

Юрий Литвинов  
y.litvinov@spbu.ru

27.03.2025

# «Стратегическое» планирование

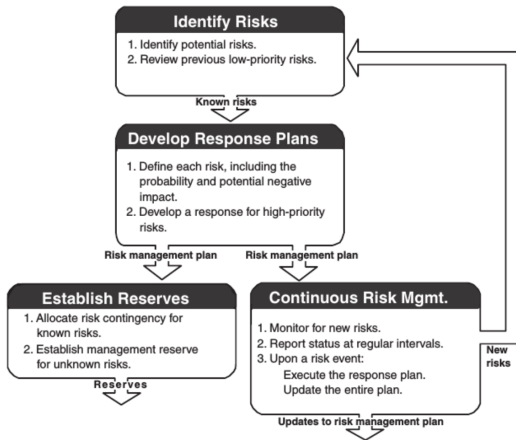
План нужен даже в Agile — это основной инструмент оценки

- ▶ Что?
- ▶ Почему/зачем?
- ▶ Когда?
- ▶ Как?
- ▶ Где?
- ▶ Кто?



# Управление рисками

- ▶ Риск =  
неопределённость +  
неприятный исход
  - ▶ Известное  
неизвестное
  - ▶ Неизвестное  
неизвестное
- ▶ Управление  
рисками —  
первоочередная  
задача менеджера  
проекта



## Шаг 1: идентификация рисков

- ▶ Получение информации от stakeholder'ов
  - ▶ Мозговые штурмы
  - ▶ Интервью
- ▶ Использование прошлого опыта
  - ▶ Построение профиля рисков
  - ▶ анализ аналогичных проектов
- ▶ Риски графика работ и бюджета

## Шаг 2: разработка стратегии противодействия

- ▶ Определение серьёзности риска
  - ▶ Описание условий возникновения и последствий
- ▶ Определение вероятности возникновения
  - ▶ Количественные оценки
  - ▶ Субъективная качественная оценка
- ▶ Определение стратегии снижения возможного урона
  - ▶ Принять
  - ▶ Избежать
  - ▶ Переложить на кого-то другого
  - ▶ Смягчить
  - ▶ Отслеживать
    - ▶ События-триггеры

Impact	Risk Management Actions		
Significant	Considerable management required	Must manage and monitor risks	Extensive management essential
Moderate	Risks may be worth accepting with monitoring	Management effort worthwhile	Management effort required
Minor	Accept risks	Accept, but monitor risks	Manage and monitor risks
	Low	Medium	High
	Likelihood		

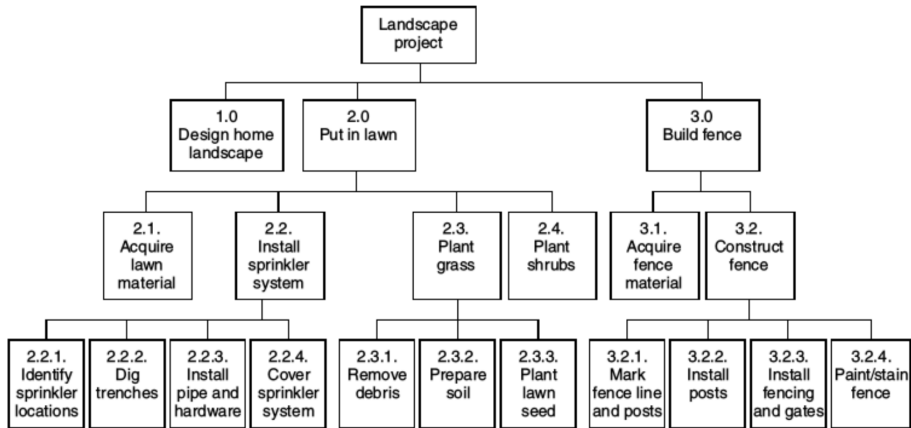
## Шаг 3: Создать резервный фонд

- ▶ Противодействие известным рискам
  - ▶ Определить План Б для каждого отслеживаемого риска
  - ▶ Оценить стоимость Плана Б для каждого риска
  - ▶ Умножить на вероятности и сложить
  - ▶ Согласовать «разумный» бюджет
- ▶ А ещё есть неизвестное неизвестное!
  - ▶ +5-30% бюджета в зависимости от типа проекта

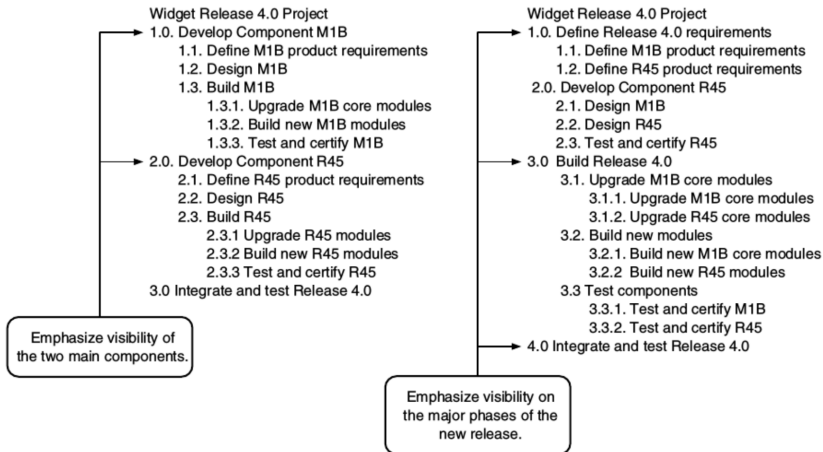
## Шаг 4: Непрерывное управление рисками

- ▶ Поддержание списка рисков
- ▶ Запланированные переоценки известных рисков
- ▶ Поиск информации, поиск новых рисков
- ▶ Анализ резервного фонда

# Декомпозиция проекта







## Критерии хорошей декомпозиции

- ▶ Не TODO — требуются отчуждаемые артефакты
- ▶ Полнота разбиения задач
  - ▶ На всех уровнях сумма объёма дочерних работ в точности равна объёму работы дочернего узла
  - ▶ Отдельные работы друг с другом не пересекаются
- ▶ Понятность и конкретность задач
  - ▶ Явный вид деятельности
  - ▶ Явный результат

# Критерии SMART

- ▶ Specific — задача должна быть конкретной
  - ▶ И однозначно пониматься всеми участниками
- ▶ Measurable — задача должна быть измеримой
  - ▶ KPI, желательно числовые
- ▶ Achievable — задача должна быть достижимой
  - ▶ Реальные сроки
  - ▶ Опирается на объективные показатели (предыдущий опыт, средние показатели)
- ▶ Relevant — задача должна быть значимой
  - ▶ Укладываться в общую стратегию проекта
- ▶ Time bound — задача должна быть ограниченной по времени
  - ▶ Правило 8/80

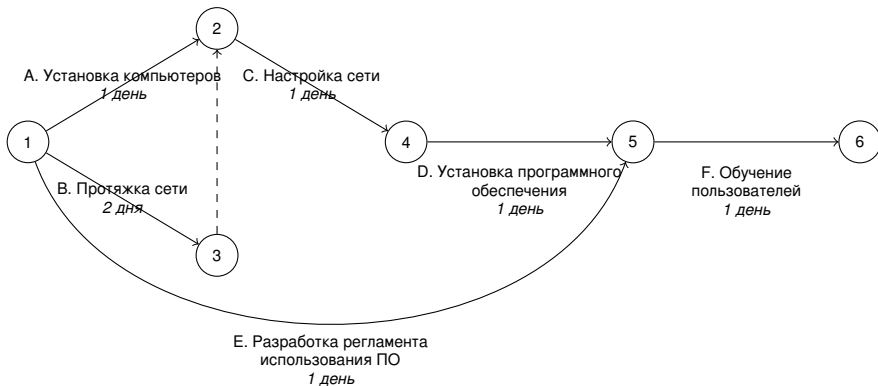
# Планирование проекта

1. Определение целей проекта и требований
2. Анализ рисков
3. Декомпозиция проекта
4. Выявление зависимостей между задачами
5. Оценка задач
  - ▶ Методом «сверху вниз» и «снизу вверх»
6. Создание и оценка плана работ
7. Распределение и оптимизация ресурсов

# Матрица зависимостей

Операция	Непосредственно предшествующие операции	Длительность
А. Установка компьютеров	—	1
В. Протяжка сети	—	2
С. Настройка сети	А, В	3
Д. Установка программного обеспечения	С	1
Е. Разработка регламента использования ПО	—	4
Ф. Обучение пользователей	Д, Е	3

# Сетевой график



# Оценка задач

## 1. Длительность

- ▶ Календарное время от начала работ до получения конечного результата
- ▶ Часы, дни, ...

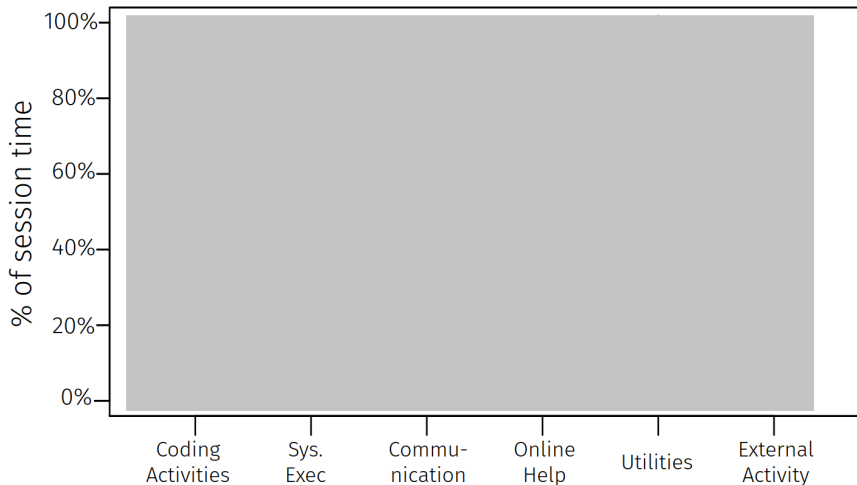
## 2. Объём работ

- ▶ Абстрактные единицы работы для решения задачи
- ▶ Человеко-часы, человеко-дни, story points, ...

## 3. Конвертация одного в другое

$$\text{Длительность работ} = \frac{\text{Объём работ}}{\text{Производительность}}$$

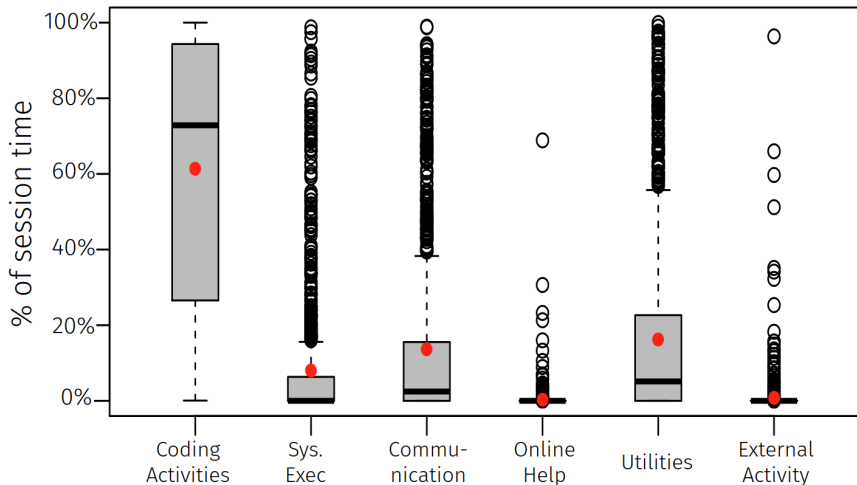
## Чем занимаются программисты, когда пишут код



© Astromskis et al. Patterns of Developers Behaviour: A 1,000-hour Industrial Study, 2017



# Чем занимаются программисты, когда пишут код



© Astromskis et al. Patterns of Developers Behaviour: A 1,000-hour Industrial Study, 2017

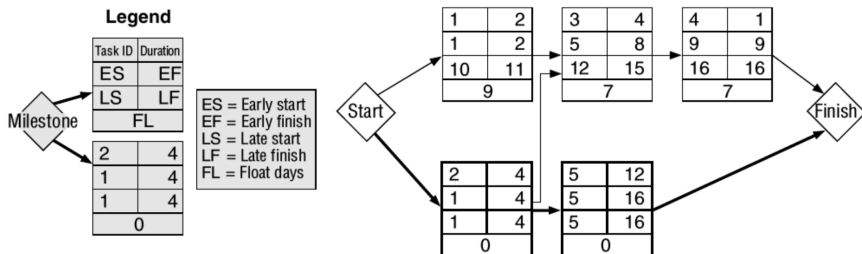
# Чем занимаются программисты на работе вообще

Activity Category		% of time over whole period
Development		
Coding	reading/editing/navigating code (and other code related activities)	21.0%
Debugger Use	using the debugger inside the IDE	0.4%
Code Reviews	performing code reviews	1.3%
Version Control	reading/accepting/submitting changes	0.7%
Email	reading/writing emails	14.5%
Planning	editing work items/tasks/todos; creating/changing calendar entries	4.8%
Read/write documents	reading/editing documents and other artifacts, e.g. pictures	6.6%
Planned meeting	scheduled meeting/call	6.5%
Informal meeting	ad-hoc, informal communication; e.g. unscheduled phone call / IM, or colleague asks a question	3.4%
Work related browsing	Internet browsing related to code/work/task	11.4%
Work unrelated browsing	Internet browsing work unrelated	5.9%
Other	Anything else; aggregates several small sub-categories, such as changing music, updating software, using the file explorer or having a break	11.4%
Other RDP	Remotedesktop use which could not be mapped to another category	12.0%

© Meyer et al. The work life of developers: Activities, switches and perceived productivity, 2017

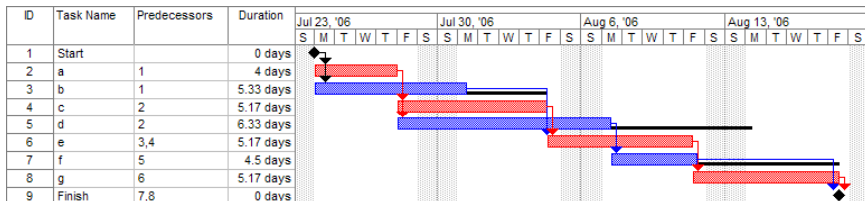
# Оценка графика работ

- ▶ Прямой проход
- ▶ Обратный проход
- ▶ Вычисление резервов
- ▶ Критический путь

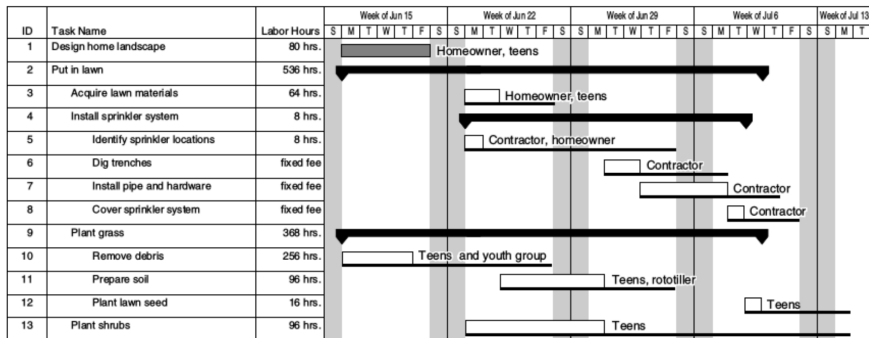


# Диаграмма Ганта

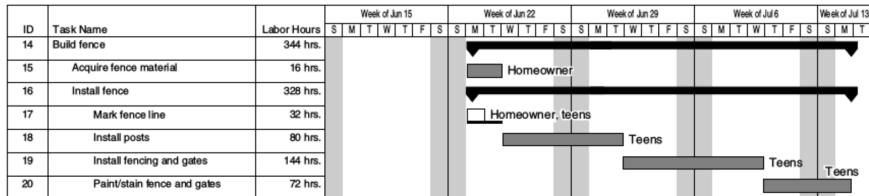
- ▶ 1910 год!
- ▶ Календарный график + зависимости работ
- ▶ Early start



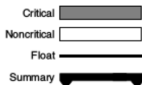
# Диаграмма Ганта с ресурсами



# Загруженность ресурсов



## Legend

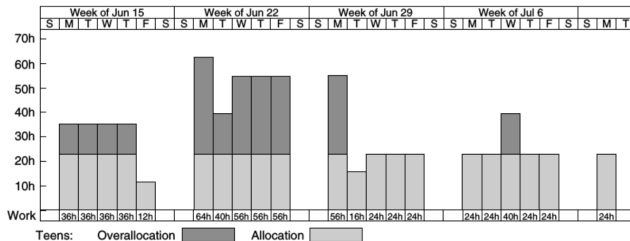


ID	Resource Name	Week of Jun 15							Week of Jun 22							Week of Jun 29							Week of Jul 6							Week of Jul 13			
		S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	
1	Homeowner		4	4	4	4	4			32	16																						
2	Teens		36	36	36	36	12			64	40	56	56	56			56	16	24	24	24				24	24	40	24	24			24	
3	Contractor									8								8	8	8	8				8	8							
4	Youth group		40	40	40	40																											
5	Rototiller										8	8	8				8																

Hours per day

# Оптимизация ресурсов

- ▶ Перегруженность и недозагруженность
- ▶ Оценка ресурсов по начальному графику
- ▶ Определение и выравнивание пиков
- ▶ Переоценка задач, перераспределение людей



# Планирование денежного потока

Resource spreadsheet

		Week of Jun 15							Week of Jun 22							Week of Jun 29							Week of Jul 6							Week of Jul 13							Week of Jul 20							Week of Jul 27								
ID	Resource Name	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S		
1	Homeowner	4	4	4	4	4			8	8						8		8	8					8																												
2	Teens	24	24	24	24	24			24	24	24	24	24			24	24	24	24	24				24	24	24	24			16	16	24	24	24			24	24	24	24				24								
3	Contractor															8	8	8	8	8				8	8																											
4	Youth group	40	40	40	40	40																																														
5	Rototiller															8	8	8					8																													

Cash flow schedule

ID	Resource Name	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7
1	Labor	\$3,400	\$1,360	\$1,440	\$1,040	\$1,040	\$1,200	\$240
2	Contractor	\$0	\$0	\$0	\$2,200	\$0	\$0	\$0
3	Equipment	\$0	\$120	\$40	\$0	\$0	\$0	\$0
4	Materials	\$0	\$1,000	\$2,000	\$0	\$0	\$0	\$0
5	Weekly total	\$3,400	\$2,480	\$3,480	\$3,240	\$1,040	\$1,200	\$240

Hours per day



# Типичные ошибки при оценке проектов

- ▶ Оценку делали не те люди
  - ▶ Мало опыта, непонимание техник оценивания
- ▶ Слишком быстрый ответ
  - ▶ Оценка в условиях недостаточной информации
- ▶ Забыли про риски и прочие буферы
- ▶ Забыли налоги
- ▶ Забыли про расходы на «административный аппарат»
- ▶ Забыли про отпуск
- ▶ Забыли про индексацию зарплат
- ▶ Забыли про закупки
- ▶ Политика vs здравый смысл

# Уровни детальности оценки

- ▶ «Оценка в лифте»
- ▶ Оценка при выборе проекта
- ▶ Детальная оценка

