Лекция 17
БГТУ, ФИТ, 1 семестр

Основы программной инженерии (ПИ)

Технологии разработки ПО. Управление проектом

План лекции:
- управление проектом;
- управление командой проекта.

На прошлых лекциях:

1. Методология гибкой разработки программного обеспечения Agile

Agile – семейство гибких итеративно-инкрементальных методов управления проектами и продуктами.

Agile-манифест – это документ, в котором излагаются ключевые ценности и принципы, лежащие в основе Agile-философии, и который призван помочь командам разработчиков работать более эффективно и устойчиво

Идеи Agile

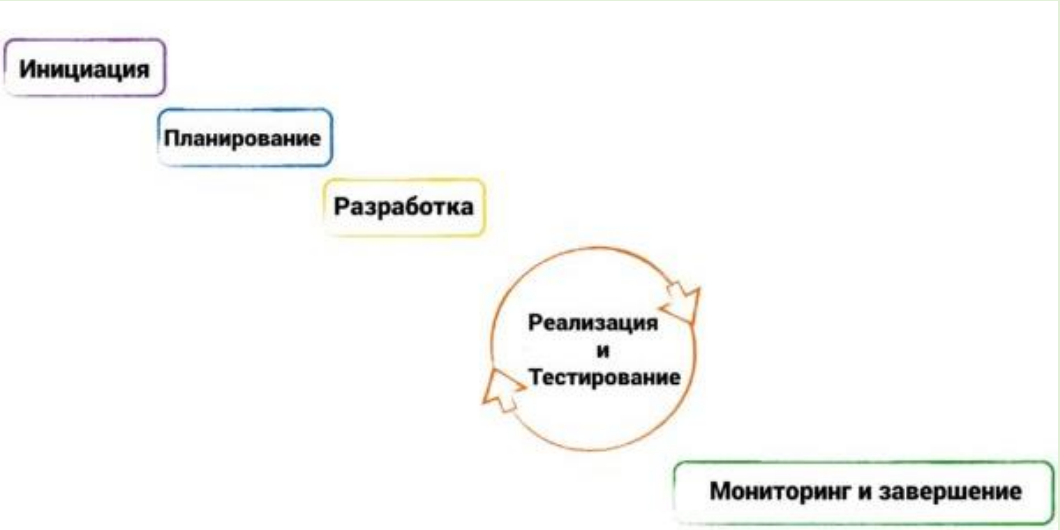
- ✓ Люди и их ***взаимодействие*** важнее, чем процессы и инструменты
- ✓ ***Рабочее*** ПО важнее, чем документация
- ✓ ***Клиенты и сотрудничество*** с ними важнее, чем контракт и обсуждение условий
- ✓ *Готовность к* ***внесению изменений*** важнее, чем первоначальный план

Принципы Agile

- ✓ ***Удовлетворять клиентов, заблаговременно и постоянно поставляя ПО (клиенты довольны, когда рабочее ПО поступает к ним регулярно и через одинаковые промежутки времени)***
- ✓ ***Изменять требования к конечному продукту в течение всего цикла его разработки***
- ✓ ***Поставлять рабочее ПО как можно чаще (раз в неделю, в две недели, в месяц и т.д.)***
- ✓ ***Поддерживать сотрудничество между разработчиками и заказчиком в течение всего цикла разработки***
- ✓ ***Поддерживать и мотивировать всех, кто вовлечен в проект (если команда мотивирована, она намного лучше справляется со своими задачами, нежели команда, члены которой условиями труда недовольны)***
- ✓ ***Обеспечивать непосредственное взаимодействие между разработчиками (возможность прямого контакта способствует более успешной коммуникации)***
- ✓ ***Измерять прогресс только посредством рабочего ПО (клиенты должны получать только функциональное и рабочее программное обеспечение)***
- ✓ ***Поддерживать непрерывный темп работы (команда должна выработать оптимальную и поддерживаемую скорость работы)***
- ✓ ***Уделять внимание дизайну и техническим деталям (благодаря эффективным навыкам и хорошему дизайну команда проекта получает возможность постоянного совершенствования продукта и работы над его улучшением)***
- ✓ ***Стараться сделать рабочий процесс максимально простым, а ПО – простым и понятным***
- ✓ ***Позволять членам команды самостоятельно принимать решения (если разработчики могут сами принимать решения, самоорганизовываться и общаться с другими членами коллектива, обмениваясь с ними идеями, вероятность создания качественного продукта существенно возрастает)***
- ✓ ***Постоянно адаптироваться к меняющейся среде (благодаря этому конечный продукт будет более конкурентоспособен)***

Управление проектами – это управление и организация процесса разработки ПО для достижения поставленной цели в установленное время и в рамках бюджета.

Классическое управление проектами – это метод управления проектами, основанный на водопадной модели ЖЦ ПО (Waterfall).



Подходит для типовых проектов:

- проекты с большим количеством задач;
- взаимосвязей;
- дедлайнов;
- видов ресурсов.

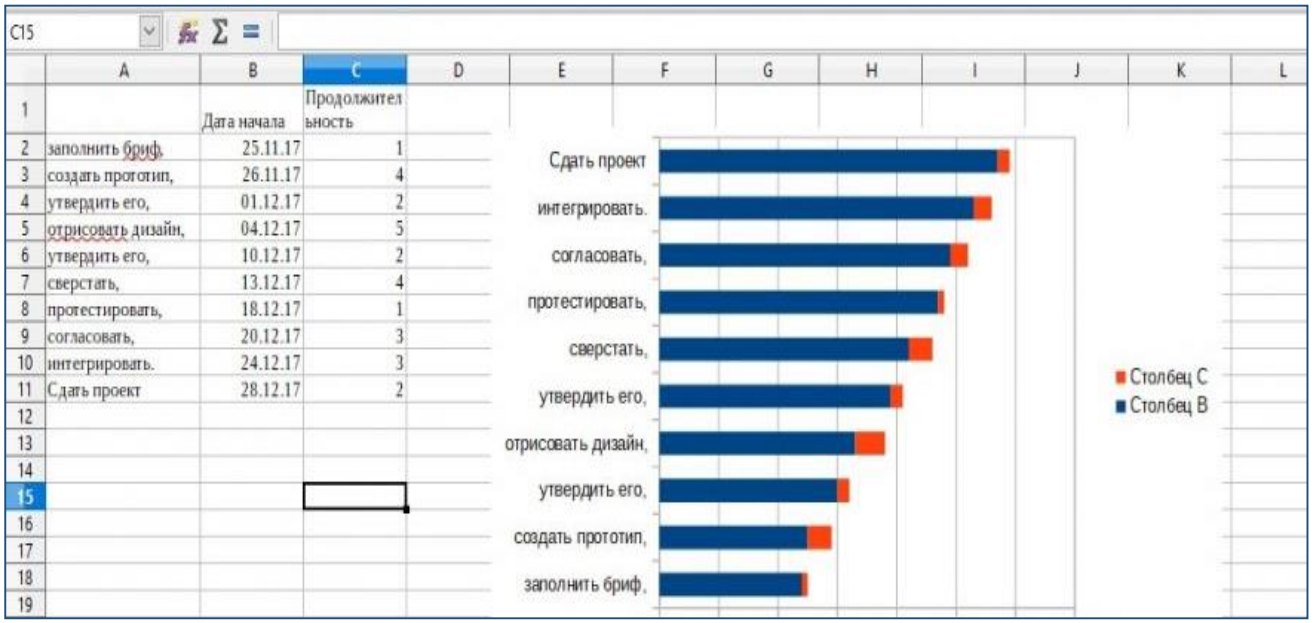
Простейший способ планирования реализации проекта –
разбить его на фазы или отдельные задачи.

Простейший инструмент проектного управления –
список (чек-лист) действий, необходимых для достижения цели.

Инструменты проектного управления

Диаграмма Гантта:

- ✓ подходит для проектов с жесткими ресурсными ограничениями;
- ✓ предоставляет возможность построить схему в форме событийной цепочки процессов (event-driven process chain) для планирования ресурсов.



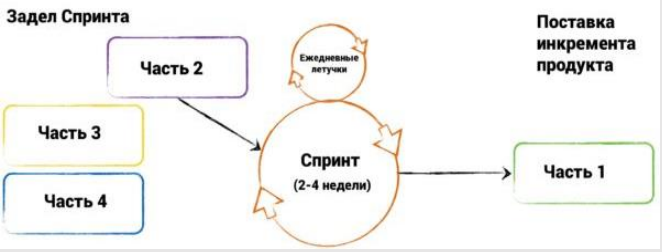
	A	B	C
1		Дата начала	Продолжительность
2	заполнить бриф,	25.11.17	1
3	создать прототип,	26.11.17	4
4	утвердить его,	01.12.17	2
5	отрисовать дизайн,	04.12.17	5
6	утвердить его,	10.12.17	2
7	сверстать,	13.12.17	4
8	протестировать,	18.12.17	1
9	согласовать,	20.12.17	3
10	интегрировать.	24.12.17	3
11	Сдать проект	28.12.17	2

Гибкий итеративно-инкрементальный подход к управлению проектами

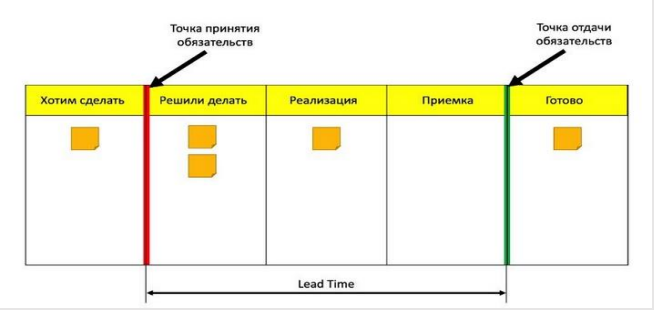
Гибкий итеративно-инкрементальный подход к управлению проектами

- ✓ ориентирован на динамическое формирование требований;
- ✓ обеспечение их реализации;
- ✓ постоянного взаимодействия внутри самоорганизующихся рабочих групп; группы состоят из специалистов различного профиля.

метод Scrum:



метод Kanban



и другие методы.

Треугольник ограничений проекта

Управление командой проекта

> ***«Из всех трудностей, с которыми столкнулись НАСА, отправляя человека на Луну, управление было наверно самой сложной задачей»***
>
> ***— Роджер Лаунис, историк НАСА***

Модель производственной архитектуры

- ✓ ориентирован на динамическое формирование требований;
- ✓ обеспечение их реализации;
- ✓ постоянного взаимодействия внутри самоорганизующихся рабочих групп; группы состоят из специалистов различного профиля.

Основные задачи:

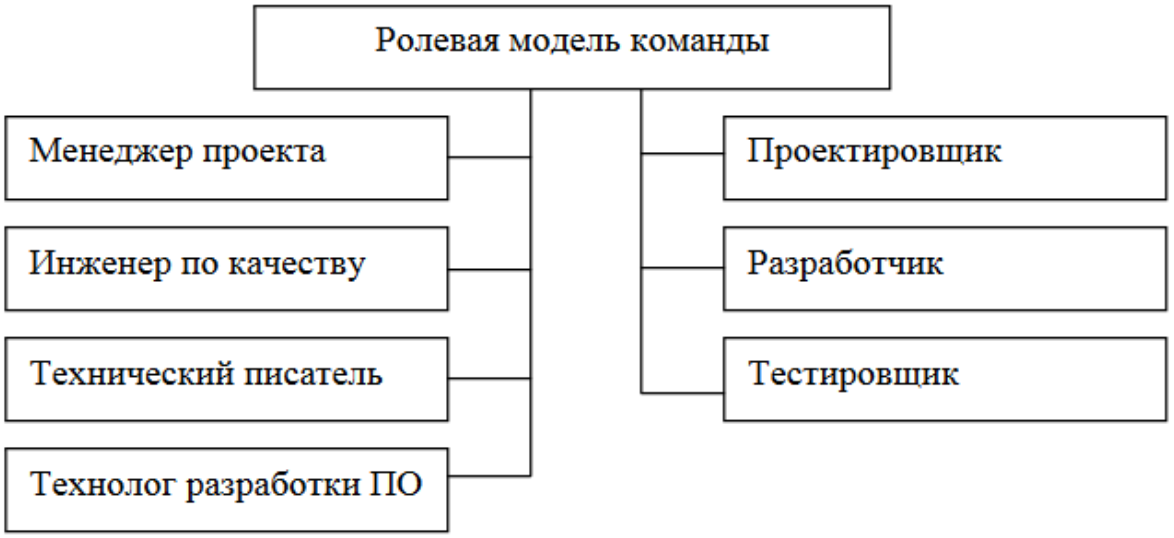
1) Подбор и управление командой.
2) Выбор процесса.
3) Выбор инструментальных средств.

Основные вопросы управления командой проекта:

- ролевая модель команды;
- модели организации команд;
- общение в команде.

Ролевая модель команды

Состав команды определяется также типом выполняемых работ:



- менеджер проекта – главное действующее лицо:
  - подбор и управление кадрами;
  - подготовка и исполнение плана проекта;
  - руководство командой;
  - обеспечение связи между подразделениями;
  - обеспечение готовности продукта.
- проектировщик – проектирование архитектуры высокого уровня и контроль ее выполнения. В небольших командах функция распределяется между менеджером и разработчиками. В больших проектах это может быть целый отдел.
- разработчик – роль, ответственная за непосредственное создание конечного продукта.
- тестировщик – роль, ответственная за удовлетворение требований к продукту (функциональных и нефункциональных).
- инженер по качеству. Обеспечивает три уровня качества:
  - качество конечного продукта – обеспечивается тестированием;
  - качество процесса разработки (тезис: для повышения качества продукта надо повысит качество процесса разработки);
  - качество (уровень) организации работ (тезис: для повышения качества процесса надо повысить качество организации работ).
- технический писатель или разработчик пользовательской (и иной) документации как части программного продукта (включая организацию тестирования документации).
- технолог разработки ПО обеспечивает поддержку модели ЖЦ, создание и сопровождение среды сборки продукта, управление исходными текстами.

Модели организации команд

«...методологи разрабатывают сложные системы, в которых есть весьма изменчивые и нелинейные компоненты – люди» Алистэр Коуберн
Проблемы человеческого фактора: – все разные – по характеру, темпераменту, активности, целям; – все похожие – участие в проекте объединяет людей общностью целей, поиском путей достижения этих целей. – различаются по типу: • индивидуалисты – члены команды; • генераторы идей – исполнители; • ответственные – безответственные; – постоянны и изменчивы; – многообразны.

Административная модель (теория X)

Характерные черты модели:
– властная пирамида – решения принимаются сверху-вниз;
– четкое распределение ролей и обязанностей;
– четкое распределение ответственности;
– следование инструкциям, процедурам, технологиям;
– роль менеджера: планирование, контроль, принятие основных решений.

Модель хаоса (теория Y)

Характерные черты модели:
– отсутствие явно выраженных признаков власти;
– роль менеджера – поставить задачу, обеспечить ресурсами, не мешать и следить, чтобы не мешали другие;
– отсутствие инструкций и регламентированных процедур;
– индивидуальная инициатива – решения по проблеме принимается там, где проблема обнаружена;
– процесс творческий на основе дружеской соревновательности.

Открытая архитектура (теория Z)

Характерные черты модели:
– адаптация к условиям работы – если нужна помощь, то обсуждаем идеи совместно;
– коллективное обсуждение проблем –принятое решение, обязательно для всех;
– распределенная ответственность;
– динамическое изменение состава рабочих групп в зависимости от текущих задач;
– отсутствие специализации;
– задача менеджера – активное участие в процессе, контроль обсуждений, обеспечение возможности активного участия всех.

Модель проектной группы MSF for Agile Software Development

Модель проектной группы и принципы, положенные в основу построения команды.

Построение команды в MSF соответствует ряду концепций (key concepts), первые три кажутся самоочевидными, остальные являются «ноу-хау»:

- концентрация на нуждах заказчика (customer-focused mindset);
- нацеленность на конечный результат (product mindset);
- установка на отсутствие дефектов (zero-defect mindset);
- «Проектная группа – команда равных» (teem of peers);
- стремление к самосовершенствованию (willingness to learn).

Ролевые группы и роли MSF

Методология MSF for Agile Software Development выделяет 7 ролевых групп

- управление программой (program management);
- управление продуктом (product management);
- управление выпуском (release operations);
- архитектура продукта (architecture);
- разработка (development);
- тестирование (test);
- удовлетворение потребителя (user experience).

Схема модели команды в MSF 4.0 – ролевые группы:

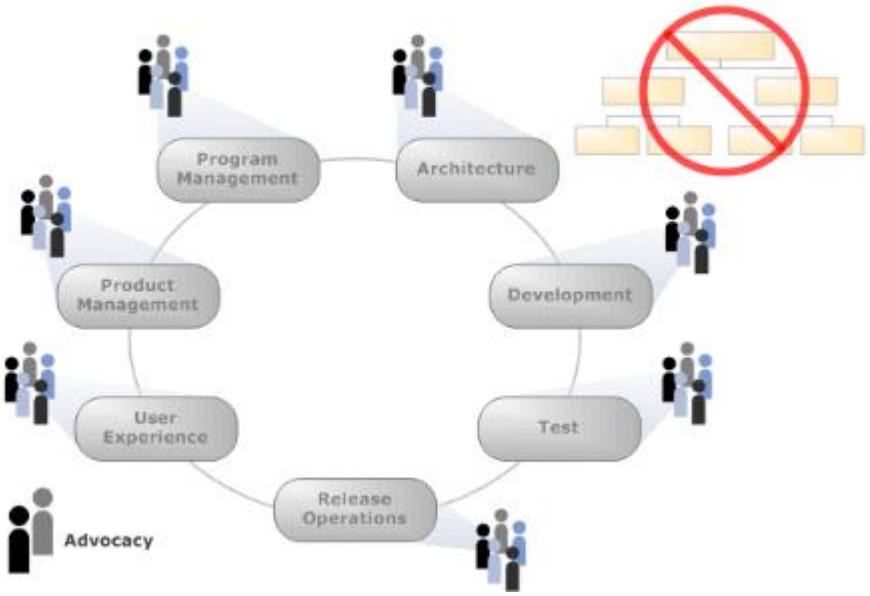
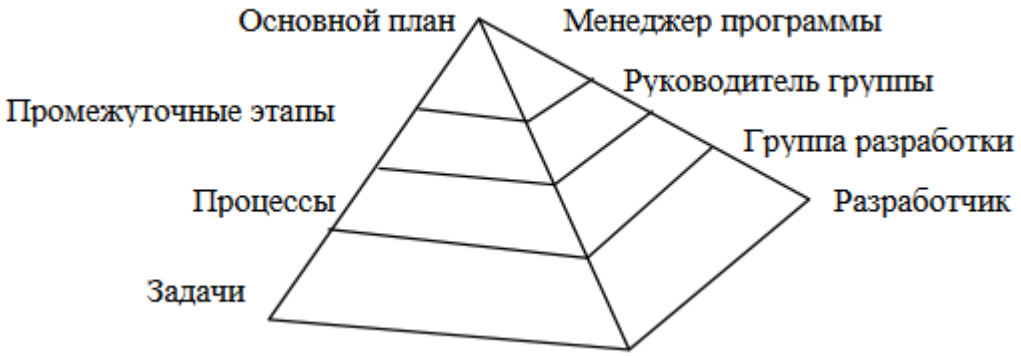


Схема роли команды в MSF 4.0 – шесть ролей:

Роли команды в MSF 4.0:

- менеджер проекта (project manager);
- бизнес-аналитик (business analyst);
- релиз-менеджер (release manager);
- архитектор (archrect);
- разработчик (developer).
- тестер (tester).

Составление графика «снизу-вверх»



Возможное объединение ролей

	Архитектура продукта	Управление продуктом	Управление программой	Разработка	Тестирование	Удовлетворение потребителя	Управление выпуском
Архитектура продукта		Нет	Да	Да	Не желательно	Не желательно	Не желательно
Управление продуктом	Нет		Нет	Нет	Да	Да	Не желательно
Управление программой	Да	Нет		Нет	Не желательно	Не желательно	Да
Разработка	Да	Нет	Нет		Нет	Нет	Нет
Тестирование	Не желательно	Да	Не желательно	Нет		Да	Да
Удовлетворение потребителя	Не желательно	Да	Не желательно	Нет	Да		Не желательно
Управление выпуском	Не желательно	Не желательно	Да	Нет	Да	Не желательно	