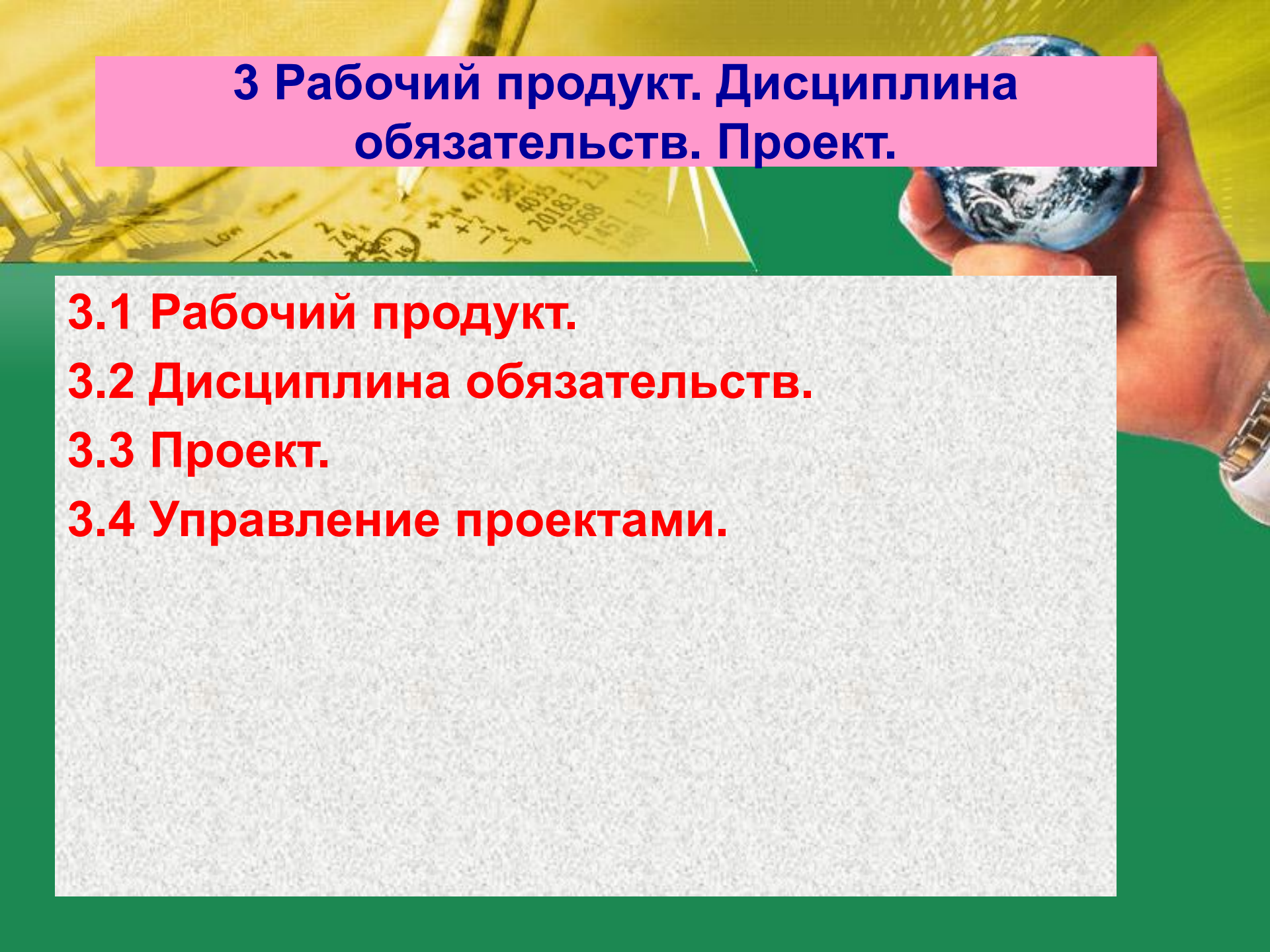




# Управление проектами

Утепбергенов Ирбулат  
Туремуратович, д.т.н., профессор

Лекция 3 «Рабочий продукт,  
дисциплина обязательств, проект»



## **3 Рабочий продукт. Дисциплина обязательств. Проект.**

**3.1 Рабочий продукт.**

**3.2 Дисциплина обязательств.**

**3.3 Проект.**

**3.4 Управление проектами.**

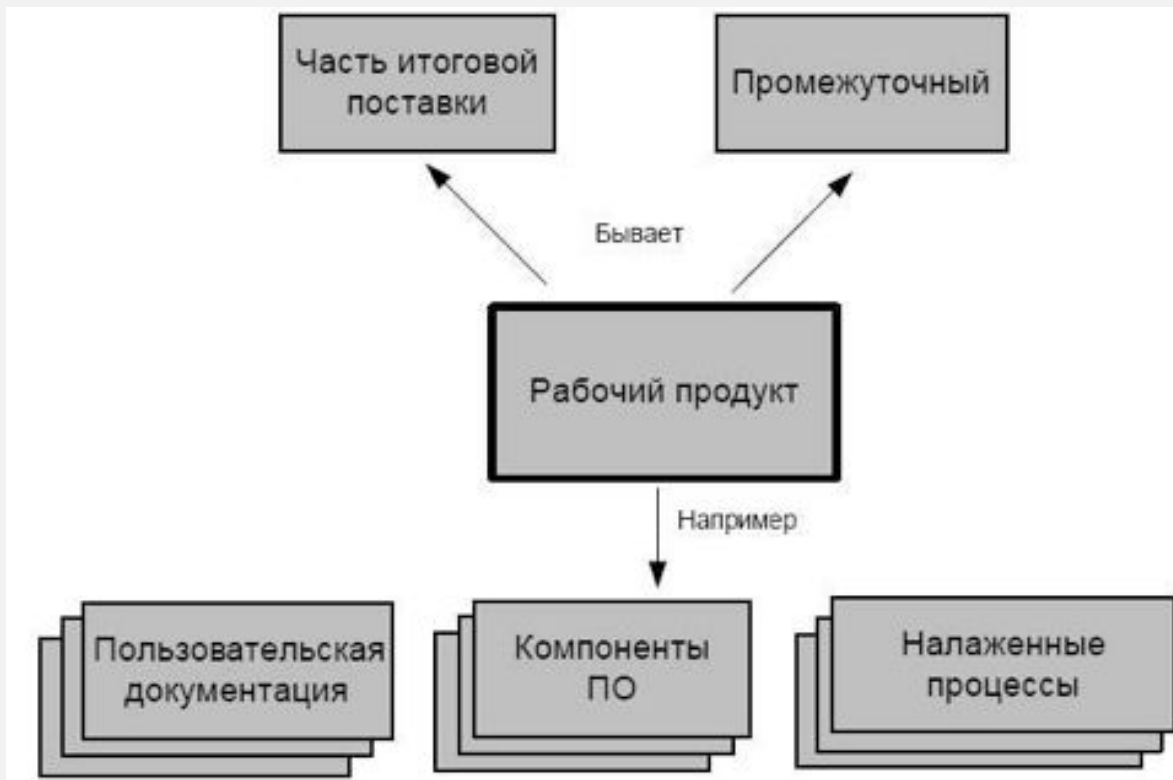


## 3.1 Рабочий продукт.

- Одним из существенных условий для управляемости промышленного процесса является наличие отдельно оформленных результатов работы – как в окончательной поставке так и промежуточных.
- Эти отдельные результаты в составе общих результатов работ помогают **идентифицировать, планировать и оценивать** различные части результата.
- Промежуточные результаты помогают **менеджерам** разных уровней отслеживать процесс воплощения проекта, **заказчик** получает возможность ознакомиться с результатами задолго до окончания проекта.
- Более того, сами участники проекта в своей ежедневной работе получают простой и эффективный способ обмена рабочей информацией – обмен результатами.

## 3.1 Рабочий продукт.

**Рабочий продукт** (*work product*) – любой *артефакт*, произведенный в процессе разработки ПО, например, файл или набор файлов, документы, составные части продукта, сервисы, процессы, спецификации, счета и т.д.





## 3.1 Рабочий продукт.

- Ключевая разница между рабочим продуктом и компонентой ПО заключается в том, что первый необязательно материален и осязаем (*not to be engineered*), хотя может быть таковым. Нематериальный *рабочий продукт* – это, как правило, некоторый налаженный процесс – промышленный процесс производства какой-либо продукции, учебный процесс в университете и т.д.
- *Рабочий продукт* совсем не обязательно является составной частью итоговой поставки.
  - Например, налаженный процесс тестирования системы не поставляется заказчику вместе с самой системой. Умение управлять проектами (не только в области программирования) во многом связано с искусством определять нужные рабочие продукты, настаивать на их создании и в их *терминах* вести приемку промежуточных этапов работы, организовывать синхронизацию различных *рабочих групп* и отдельных специалистов.
- Многие методологии включают в себя описание специфичных рабочих продуктов, используемых в процессе – *CMMI*, *MSF*, *RUP* и др. Например, в *MSF* это программный код, *диаграммы* приложений и классов (*application diagrams* и *class diagrams*), план *итераций* (*iteration plan*), *модульный тест* (*unit test*) и др. Для каждого из них точно описано содержание, ответственные за разработку, место в процессе и др. аспекты.



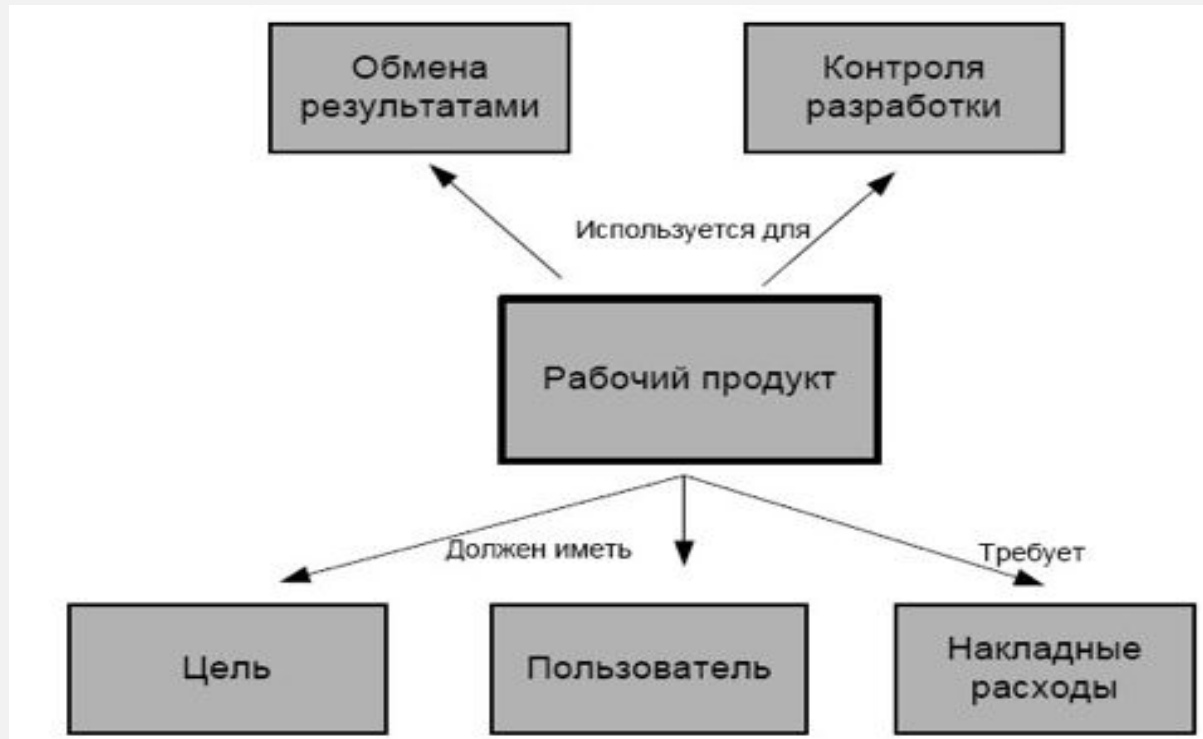
## 3.1 Рабочий продукт.

### промежуточный рабочий продукт

- Компонента ПО, созданная в проекте одним разработчиком и предоставленная для использования другому разработчику, оказывается рабочим продуктом.
  - Ее надо минимально протестировать, поправить имена интерфейсных классов и методов, быть может, убрать лишнее, не имеющее отношение к функциональности данной компоненты, разделить *public* и *private*, и т.д.
  - То есть проделать некоторую дополнительную работу, которую, быть может, разработчик и не стал делать, если бы продолжал использовать компоненту только сам.
  - Объем этих дополнительных работ существенно возрастает, если компонента должна быть представлена для использования в разработке, например, в другой центр разработки (например, иностранным партнерам, что является частой ситуацией в оффшорной разработке).
- Итак, изготовление хороших промежуточных рабочих продуктов очень важно для успешности проекта, но требует дополнительной работы от их *авторов*. Работать одному, не предоставляя рабочих продуктов – легче и для многих предпочтительнее. Но работа в команде требует накладных издержек, в том числе и в виде трат на создание промежуточных рабочих продуктов. Конечно, качество этих продуктов и трудозатраты на их изготовление сильно варьируются в зависимости от ситуации, но тут важно понимать сам принцип.

## 3.1 Рабочий продукт.

Промежуточный *рабочий продукт* должен обязательно иметь ясную цель и конкретных пользователей, чтобы минимизировать накладные расходы на его создание.





## 3.2 Дисциплина обязательств.

- В основе *разделения обязанностей* в бизнесе и промышленном производстве, корпоративных правил и *норм* лежит определенная деловая этика, форма отношений – **дисциплина обязательств**.
- Она широко используется на практике и является одной из возможных форм социального взаимоотношения между людьми.
- Привнесение в бизнес и промышленность иных моделей человеческих отношений – семейных, дружеских и т.д. часто наносит делам серьезный урон, порождает конфликтность, понижает эффективность.
- Дисциплине обязательств уделяется много внимания в рамках *MSF*, поскольку там в модели команды нет лидера, начальника. Эта дисциплина реализована также в Scrum: Scrum-команда имеет много свобод, и в силу этого – большую ответственность. Регламентируются также правила действий, когда обязательства не могут быть выполнены такой командой.



## 3.2 Дисциплина обязательств.

Основой **дисциплины обязательств** являются обязательства, которые:

- даются добровольно;
- не даются легко – работа, ресурсы, расписание должны быть тщательно учтены;
- между сторонами включает в себя то, *что* будет сделано, *кем* и в *какие сроки* ;
- открыто и публично сформулированы (то есть это не "тайное знание").

Кроме того:

- ответственная сторона стремится выполнить обязательства, даже если нужна помощь;
- до наступления *deadline*, как только становится очевидно, что работа не может быть закончена в срок, обсуждаются новые обязательства.



## 3.3 Проект.

- **Проект** – это уникальная (в отличии от традиционной пооперационного промышленного производства) деятельность, имеющая:
  - начало и конец во времени,
  - направленная на достижение определённого результата/цели,
  - создание определённого, уникального продукта или услуги, при заданных ограничениях по ресурсам и срокам,
  - а также требованиям к качеству и допустимому уровню риска.
- В частности, *разработка программного обеспечения*, является, преимущественно, проектной областью

A hand holding a small globe of the Earth against a background of architectural blueprints and a pencil.

## 3.3 Проект.

- Необходимо различать проекты промышленные и проекты творческие. У них разные принципы управления. Сложность промышленных проектов – в большом количестве разных организаций, компаний и относительной уникальности самих работ. Пример – строительство многоэтажного дома. Сюда же относятся различные международные проекты и не только промышленные – образовательные, культурные и пр. Задача в управлении такими проектами – это все охватить, все проконтролировать, ничего не забыть, все свести воедино, добиться движения, причем движения согласованного.
- Творческие проекты характеризуются абсолютной новизной идеи – новый сервис, абсолютно новый *программный продукт*, какого еще не было на рынке, проекты в области искусства и науки. Любой начинающий бизнес, как правило, является таким вот творческим проектом. Причем новизна в подобных проектах не только абсолютная – такого еще не было. Такое, может, уже и было, но только не с нами, командой проекта. То есть присутствует огромный объем относительной новизны для самих людей, которые воплощают этот проект.



## 3.3 Проект.

- Проекты по разработке *программного обеспечения* находятся между двумя этими полюсами, занимая в этом пространстве различное положение.
- Часто они сложны потому, что объемны и находятся на стыке различных дисциплин – того целевого бизнеса, куда должен встроиться *программный продукт*, и сложного, нетривиального программирования.
- Часто сюда добавляется еще разработка уникального электронно-механического оборудования.
- С другой стороны, поскольку программирование *активно* продвигается в разные сферы человеческой *деятельности*, то происходит это путем создания абсолютно новых, уникальных продуктов, и их разработка и продвижение обладают всеми чертами творческих проектов.

A hand holding a small globe of the Earth. The background is a collage of architectural blueprints, a pen, and a green surface. A pink banner is overlaid on the top part of the image.

## 3.4 Управление проектами.

- **Управление проектами** (*project management*) – область деятельности, в ходе которой, в рамках определенных проектов, определяются и достигаются четкие цели при нахождении компромисса между объемом работ, ресурсами (такими как время, деньги, труд, материалы, энергия, пространство и др.), временем, качеством и рисками.



## 3.4 Управление проектами.

аспекты управления проектами.

- **Stakeholders** – это люди со стороны, которые не участвуют непосредственно в проекте, но влияют на него и/или заинтересованы в его результатах. Это могут быть будущие пользователи системы, высшее руководство компании-разработчика и т.д. *Идентификация всех stakeholders* и грамотная работа с ними – важная составляющая успешного проектного менеджмента
- **Project scope** – это границы проекта. Это очень важное понятие для программных проектов в виду *изменчивости требований*. Часто бывает, что разработчики начинают создавать одну систему, а после, постепенно, она превращается в другую. Причем для менеджеров по продажам, а также заказчика, ничего радикально не произошло, а с точки зрения внутреннего устройства ПО, технологий, *алгоритмов* реализации, архитектуры – все радикально меняется.
- **Компромиссы** – важнейший аспект управления программными проектами в силу согласовываемости ПО. Важно не потерять все согласуемые параметры и стороны и найти приемлемый *компромисс*. Одна из техник управления *компромиссами* будет рассказана в контексте изучения методологии *MSF*.



## 3.4 Управление проектами

### Области управления проектами

- Планирование и мониторинг проекта, контроль за изменениями в проекте (*Project planning / Tracking / Change Control*)
- Управление рамками проекта (*Scope Management*)
- Управление календарным графиком проекта (*Schedule Management*)
- Управление стоимостью (*Cost Management*)
- Управление персоналом (*Staff Resource Management*)
- Управление коммуникацией (*Communications Management*)
- Управление рисками (*Risk Management*)
- Управление снабжением (*Procurement*)
- Управление качеством (*Quality Management*)



Благодарю за внимание!