Некоммерческое акционерное общество

«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»

Кафедра «IT - Инжиниринг»

**ЗАДАНИЕ №1**

По дисциплине «Теория и практика управления проектами»

На тему: Методы УП

Специальность 7M06103-Вычислительная техника и программное обеспечение

Выполнил Сакан Ерлан Группа МВТн-21-2

\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_202\_\_\_г.

(оценка) (подпись)

Алматы 2022

В зависимости от проекта данная фаза может состоять из простой передачи Заказчику результатов проекта или же из длительного процесса взаимодействия с клиентами по улучшению проекта и повышению их удовлетворённости, и поддержке результатов проекта. Последнее относится к проектам в области клиентского сервиса и программного обеспечения.

То, что описано выше – база, на которой строятся различные методы управления проектами. Разным проектам нужны различные фазы реализации – некоторым достаточно и трёх фаз, другим гораздо больше. Иногда используется так называемый «итеративный водопад», в котором каждый этап представляет собой некий подпроект, в ходе которого задачи реализуются по фиксированным итерациям. Но суть остаётся одна – проект разбит на этапы, которые исполняются в строго определённой последовательности.

Благодаря тому, что классический проектный менеджмент строго привязан ко времени исполнения задач, как правило, заранее определённому на этапе планирования, для реализации проектов в рамках данного подхода отлично подходят инструменты календарно-сетевого планирования. Самым распространённым инструментом календарно-сетевого планирования является уже упомянутая ранее диаграмма Гантта. Существует множество инструментов для её построения – от простых таблиц вроде Excel и Smartsheet до профессиональных программных пакетов вроде Microsoft Project и Primavera.

**Сильные стороны классического проектного менеджмента**

Сегодня довольно часто говорится о том, что классический водопадный подход устарел, но он и не думает сдавать позиции. Большим плюсом данного подхода является то, что он требует от Заказчика и руководства компании определить, что же они хотят получить, уже на первом этапе проекта. Раннее включение привносит определённую стабильность в работу проекта, а планирование позволяет упорядочить реализацию проекта. Кроме того, этот подход подразумевает мониторинг показателей и тестирование, что совершенно необходимо для реальных проектов различного масштаба.

Потенциально, классический подход позволяет избежать стрессов ввиду наличия запасного времени на каждом этапе, заложенного на случай каких-либо осложнений и реализации рисков. Кроме того, с правильно проведённым этапом планирования, руководитель проектов всегда знает, какими ресурсами он обладает. Даже если эта оценка не всегда точная.

**Слабые стороны классического проектного менеджмента**

Основная слабая сторона классического проектного менеджмента – нетолерантность к изменениям. Руководство компании Toyota, знаменитую созданием таких систем как Lean и Kanban, часто критикуют за то, что они применяют классический подход в разработке софта для своей компании, причём именно за недостаток гибкости.

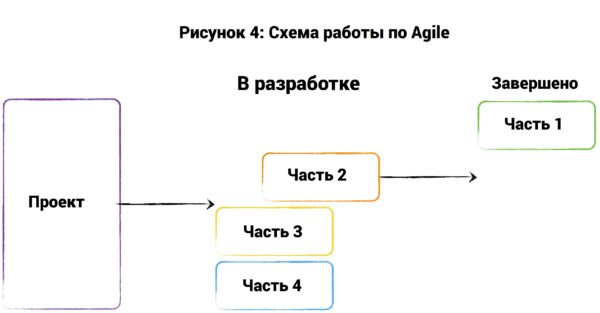
Оплот классического подхода сейчас – строительные и инженерные проекты, в которых содержание проекта остаётся практически неизменным в течение всего проекта. Но если в Вашем проекте ресурсы и время не являются ключевыми ограничениями, а содержание проекта подвержено изменениям – возможно вам стоит присмотреться к другим системам управления проектами.

Agile

Как уже говорилось ранее – не все проекты могут быть структурированы таким образом, чтобы быть реализованными по классическому проектному подходу. Возвращаясь к нашему примеру с шеф-поваром: приготовление одного блюда идеально ложится на «водопадный» подход, а вот вовремя приготовить и подать ужин из четырёх блюд будет практически невозможно, если придётся каждый раз ждать окончания приготовления одного блюда, чтобы приступить к приготовлению другого.

И тут в игру вступает Agile – семейство гибких итеративно-инкрементальных методов к управлению проектами и продуктами. Согласно данному подходу, проект разбивается не на последовательные фазы, а на маленькие подпроекты, которые затем «собираются» в готовый продукт. Схема работы приведена на *Рисунке 5*.

Таким образом, инициация и верхнеуровневое планирование проводятся для всего проекта, а последующие этапы: разработка, тестирование и прочие проводятся для каждого мини-проекта отдельно. Это позволяет передавать результаты этих мини-проектов, так называемые, инкременты, быстрее, а приступая к новому подпроекту (итарации) в него можно внести изменения без больших затрат и влияния на остальные части проекта.



Несмотря на то, что Agile вошёл в моду относительно недавно, идея итеративной разработки не нова *(об истории появления Agile можно прочесть*[*здесь*](http://www.pmservices.ru/project-management-news/neizvestnaya-istoriya-agile/)*– прим.пер.).*Своё нынешнее название семейство гибких методологий получило в 2001 с публикации Манифеста Agile (Agile Manifesto), закрепившем основные ценности и принципы гибкой разработки программного обеспечения, в основе которых – командная работа и адаптация, даже «любовь» к изменениям.

Сам по себе Agile – не метод управления проектами. Это скорее набор идей и принципов того, как нужно реализовывать проекты. Уже на основе этих принципов и лучших практик были разработаны отдельные гибкие методы или, как их иногда называют, фреймворки (frameworks): Scrum, Kanban, Crystal, и многие другие. Эти методы могут достаточно сильно отличаться друг от друга, но они следуют одним и тем же принципам.

**Сильные стороны Agile**

Самое главное достоинство Agile – его гибкость и адаптивность. Он может подстроиться под практически любые условия и процессы организации. Именно это обуславливает его нынешнюю популярность и то, сколько систем для различных областей было создано на его основе.

Один из принципов Agile: «Реакция на изменения важнее следования плану». Именно быстрая и относительно безболезненная реакция на изменения является причиной тому, что многие крупные компании стремятся сделать свои процессы более гибкими. Кроме того, Agile отлично подходит для проектов с «открытым концом» — например, запуску сервиса или блога.

Вотчина  Agile – разработка новых, инновационных продуктов. В проектах по разработке таких продуктов высока доля неопределённости, а информация о продукте раскрывается по ходу проекта. В таких условиях реализовывать проект по «водопаду» становится невозможно– нет информации для планирования.

**Слабые стороны Agile**

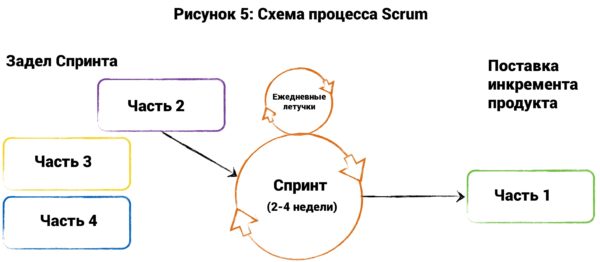
В отличие от PRINCE2 и PMBOK Agile – не является ни методологией, ни стандартом. Agile — это набор принципов и ценностей. Слабая сторона состоит в том, что каждой команде придётся самостоятельно составлять свою систему управления, руководствуясь принципами Agile. Это непростой и длительный процесс, который потребует изменений всей организации, начиная процедурами и заканчивая базовыми ценностями. Это тернистый путь и не всем организациям он под силу.

Этот путь потребует от лидера изменений не только знаний и упорства, но и серьёзных административных ресурсов, а также затрат. К счастью, существуют готовые наборы практик, которые облегчают Agile-трансформацию организации. К таким наборам относятся фреймворк Scrum, метод Kanban и многие другие – Crystal, LeSS, SAFe, Nexus.

Scrum

Гибкий фреймворк, созданный в 1986 году, считается самым структурированным из семейства Agile. Созданный в 1986 году, он сочетает в себе элементы классического процесса и идеи гибкого подхода к управлению проектами. В итоге получилось очень сбалансированное сочетание гибкости и структурированности.

Следуя заветам Agile, Scrum разбивает проект на части, которые сразу могут быть использованы Заказчиком для получения ценности, называемые заделами продуктов (product backlog). И несмотря на то, что «задел продукта» — достаточно верный перевод и используется в профессиональной литературе, в российской практике чаще всего используется просто «беклог». Затем эти части приоретизируются Владельцем продукта – представителем Заказчика в команде. Самые важные «кусочки» первыми отбираются для выполнения в Спринте – так называются итерации в Scrum, длящиеся от 2 до 4 недель.  В конце Спринта Заказчику представляется рабочий инкремент продукта – те самые важные «кусочки», которые уже можно использовать. Например, сайт с частью функционала или программа, которая уже работает, пусть и частично. После этого команда проекта приступает к следующему Спринту. Длительность у Спринта фиксированная, но команда выбирает её самостоятельно в начале проекта, исходя из проекта и собственной производительности.



Чтобы удостовериться в том, что проект отвечает требованиям Заказчика, которые имеют свойство изменяться со временем, перед началом каждого Спринта происходит переоценка ещё не выполненного содержания проекта и внесение в него изменений. В этом процессе участвуют все – команда проекта, Scrum Мастер (Scrum Master, лидер команды проекта) и Владелец продукта. И ответственность за этот процесс лежит на всех.

Как уже говорилось, Владелец продукта является представителем Заказчика в проекте, или олицетворяет всех клиентов будущего проекта, в случае если Заказчика нет. Для этого он должен досконально знать их потребности и образ мышления, а также разбираться в продукте и технологии его изготовления. Scrum Мастер призван помочь участникам проекта лучше понять и принять ценности, принципы и нормы практики Scrum. Он лидер и посредник между внешним миром и командой. Его задача — следить, чтобы никто не мешал команде самостоятельно и комфортно работать над поставленными задачами. Команда же отвечает за то, чтобы в конце спринта все необходимые задачи были сделаны, а поставки – выполнены.

Основная структура процессов Scrum вращается вокруг 5 основных встреч: упорядочивания беклога, планирования Спринта, ежедневных летучек, подведения итогов Спринта и ретроспективы Спринта.

* **Встреча по упорядочиванию беклога (Backlog Refinement Meeting, «Backlog Grooming»):** Эта встреча аналогична фазе планирования в классическом проектном управлении, и проводится в первый день каждого Спринта. На ней рассматривается – что уже было сделано по проекту в целом, что ещё осталось сделать и принимается решение о том, что же делать дальше. Владелец продукта определяет, какие задачи на данном этапе являются наиболее приоритетными. Данный процесс определяет эффективность Спринта, ведь именно от него него зависит, какую ценность получит Заказчик по итогам спринта.
* **Планирование Спринта:** После того, как Владелец продукта определил приоритеты, команда совместно решает, что же конкретно они будут делать во время грядущей итерации, как достигнуть поставленной на предыдущей встрече цели. Команды могут применять различные инструменты планирования и оценки на данном этапе, лишь бы они не противоречили принципам и логике Scrum. Планирование Спринта проводится в самом начале итерации, после Встречи по упорядочиванию продукта.
* **Ежедневные летучки:** Каждый день спринта, в идеале, в одно и то же время, члены команды тратят 15 минут на то, чтобы поделиться информацией о статусе задач и состоянии проекта. На ней не происходит обсуждений проблем или принятия решений – если после встречи возникают вопросы и конфликты, Scrum Мастер и вовлечённые участники обсуждают их отдельно. Летучка же нужна для обмена информации и поддержания всех членов команды в курсе состояния проекта.
* **Подведение итогов Спринта:** Цель этапа – обследование и адаптация создаваемого продукта. Команда представляет результаты деятельности всем заинтересованным лицам. Основная задача – убедиться, что продукт этапа соответствует ожиданиям участников и согласуется с целями проекта.
* **Ретроспектива Спринта:** Проводится сразу после Подведения итогов спринта и до планирования следующего спринта. На нём команда выясняет, насколько чётко и слаженно проходил процесс реализации этапа. Обследованию подвергаются возникшие проблемы в работе, методологии и взаимодействии. Именно этот этап позволяет команде провести рефлексию и следующий Спринт провести эффективнее.

Многим Scrum может показаться сложным для внедрения – новый процесс, новые роли, много делегирования и совершенно новая организационная структура. Но это гибкий и при этом структурированный подход к реализации проектов, который, в отличие от размытых и общих принципов Agile, не позволит работе пойти не в то русло.

**Сильные стороны Scrum**

Scrum был разработан для проектов, в которых необходимы «быстрые победы» в сочетании с толерантностью к изменениям. Кроме того, этот фреймворк подходит для ситуаций, когда не все члены команды имеют достаточный опыт в той сфере, в которой реализуется проект – постоянные коммуникации между членами командами позволяют недостаток опыта или квалификации одних сотрудников за счёт информации и помощи от коллег.

Онлайн телеканал Netflix является отличным примером быстрых поставок результатов. Сайт ресурса обновляется каждые две недели благодаря Scrum, который не просто позволяет работать с высокой скорости, но и аккумулирует пользовательский опыт и даёт возможность выявить самое главное для клиентов.

В ходе каждой итерации, разработчики добавляют и тестируют новые функции сайта и убирают те, которыми не пользовались клиенты. По словам команды Netflix, основное преимущество Scrum в том, что он позволяет «быстро ошибаться». Вместо того, чтобы долго и с большими затратами готовить крупный релиз, поставки раз в две недели по Scrum имеют небольшой размер. Их легко отслеживать и, если что-то идёт не так, быстро исправлять.

**Слабые стороны Scrum**

Scrum очень требователен к команде проекта. Она должна быть небольшой (5-9 человек) и кроссфункциональной – то есть члены команды должны обладать более чем одной компетенцией, необходимой для реализации проекта. Например разработчик ПО должен обладать познаниями в тестировании и бизнес-аналитике. Делается это для того, чтобы часть команды не «простаивала» на разных этапах проекта, а также для того, чтобы сотрудники могли помогать и подменять друг друга.

Кроме того, члены команды должны быть «командными игроками», активно брать на себя ответственность и уметь самоорганизовываться. Подобрать такую зрелую  команду очень непросто!

Scrum подходит не для всех команд и организаций ещё и потому, что предлагаемый процесс может не подойти для разработки конкретного продукта – например промышленного станка или постройки здания.

 Lean

Agile говорит нам, что необходимо разбивать на небольшие управляемые пакеты работ, но ничего не говорит о том, как управлять разработкой этого пакета. Scrum предлагает нам свои процессы и процедуры. Lean же, в свою очередь, добавляет к принципам Agile схему потока операций (workflow) для того, чтобы каждая из итераций выполнялась одинаково качественно.

В Lean, так же, как и в Scrum, работа разбивается на небольшие пакеты поставки, которые реализуются отдельно и независимо. Но в Lean для разработки каждого пакета поставки существует поток операций с этапами, подобными тем, которые были созданы для проекта Аполлон. Как и в классическом проектном менеджменте, это могут быть этапы планирования, разработки, производства, тестирования и поставки – или любые другие необходимые для качественной реализации проектов этапы.



Этапы Lean и их гибкость позволяют быть уверенными в том, что каждая часть проекта реализуется так, как требуется. В Lean не прописаны чёткие границы этапов, как в Scrum прописаны ограничения Спринтов. Кроме того, в отличие от классического проектного менеджмента, Lean позволяет параллельно выполнять несколько задач на разных этапах, что повышает гибкость и увеличивает скорость исполнения проектов.

Как и Agile, Lean это скорее концепция, образ мышления, нежели нечто высеченное в камне. Используя идеи Lean Вы можете самостоятельно создать систему, удовлетворяющую вашим требованиям в управлении проектами.

**Сильные стороны Lean**

Если Вам нравятся идеи Agile, но проект требует очень ровного качества и чёткого исполнения, Lean предоставляет набор инструментов для того, чтобы удовлетворить эти требования. Lean сочетает гибкость и структурированность, как Scrum, но в немного другом ключе.

**Слабые стороны Lean**

Не каждая часть проекта требует одинаково детальной и дотошной проработки и внимания. Но Lean предполагает именно такой подход к каждой задаче и этапу. Это основной минус применения Lean для крупных и неоднородных проектов.

А ещё, в отличие от Scrum, Lean не предлагает чёткого рабочего процесса для реализации «кусочков» проекта, что способствует растягиванию сроков проекта. Эта проблема может быть решена при помощи эффективного руководства и чётких коммуникаций  ̶  главное помнить об этом.

Kanban

Lean выглядит немного абстрактным сам по себе, но в комбинации с Kanban его становится гораздо проще использовать для построения собственной системы управления проектами. Созданный инженером компании Toyota Тайичи Оно (Taiichi Ono) в 1953 году, Kanban очень похож на схему промышленного производства. На входе в этот процесс попадает кусочек металла, а на выходе получается готовая деталь. Также и в Kanban, инкремент продукта передаётся вперёд с этапа на этап, а в конце получается готовый к поставке элемент.

Кроме того, создатель Kanban вдохновлялся супермаркетами, а именно их принципом – «держи на полках только то, что нужно клиенту». А потому в Kanban разрешается оставить неоконченную задачу на одном из этапов, если её приоритет изменился и есть другие срочные задачи. Неотредактированная статья для блога, подвешенная без даты публикации или часть кода функции, которую возможно не будут включать в продукт – всё это нормально для работы по Kanban.



Kanban намного менее строгий, нежели Scrum – он не ограничивает время спринтов, нет ролей, за исключением владельца продукта. Kanban даже позволяет члену команды вести несколько задач одновременно, чего не позволяет Scrum. Также никак не регламентированы встречи по статусу проекта – можно делать это как Вам удобно, а можно не делать вообще.

Для работы с Kanban необходимо определить этапы потока операций (workflow). В Kanban они изображаются как столбцы, а задачи обозначают специальные карточки. Карточка перемещается по этапам, подобно детали на заводе, переходящей от станка к станку, и на каждом этапе процент завершения становится выше. На выходе мы получаем готовый к поставке заказчику элемент продукта. Доска со столбцами и карточками может быть как настоящей, так и электронной – даже здесь Kanban не накладывает никаких ограничений на пользователей.

Ваша собственная система Kanban может быть настолько гибкой, насколько Вы сами того пожелаете – ведь во многом Kanban является визуализацией идеи Agile. Но у Kanban есть 4 столпа, на которых держится вся система:

1. **Карточки:** Для каждой задачи создаётся индивидуальная карточка, в которую заносится вся необходима информация о задаче. Таким образом, вся нужная информация о задаче всегда под рукой.
2. **Ограничение на количество задач на этапе:** Количество карточек на одном этапе строго регламентировано. Благодаря этому сразу становится видно, когда в потоке операций возникает «затор», который оперативно устраняется.
3. **Непрерывный поток:** Задачи из беклога попадают в поток в порядке приоритета. Таким образом, работа никогда не прекращается.
4. **Постоянное улучшение («кайзен» (kaizen)):** Концепция постоянного улучшения появилась в Японии в конце XX века. Её суть в постоянном анализе производственного процесса и поиске путей повышения производительности.

**Сильные стороны Kanban**

Как и Scrum, Kanban хорошо подходит для достаточно сплочённых команды с хорошей коммуникацией. Но в отличие от Scrum, в Kanban нет установленных чётких дедлайнов, что хорошо подходит для замотивированных и опытных команд.

При правильной настройке и управлении, Kanban может принести большую пользу команде проекта. Точный расчёт нагрузки на команду, правильная расстановка ограничений и концентрация на постоянном улучшении — всё это позволяет Kanban серьёзно экономить ресурсы и укладывать в дедлайны и бюджет. И всё это в сочетании с гибкостью.

**Слабые стороны Kanban**

Часто можно слышать, что по Kanban, в отличие от Scrum, можно работать с практически любой командой. Но это не совсем так. Kanban лучше всего подходит для команд, навыки членов которых пересекаются друг с другом. Таким образом они могут помогать друг другу преодолевать трудности при решении задач. Без этого Kanban будет не так эффективен, как мог бы быть. Также, как уже было сказано, Kanban лучше подходит в тех случаях, когда нет жёстких дедлайнов. Для жёстких дедлайнов лучше подходит классический подход или Scrum.

6 сигм (Six Sigma)

Компания Motorola, наряду с Toyota, также внесла вклад в развитие мирового проектного управления. Инженер этой компании Bill Smith создал концепцию 6 сигм в 1986 году. Это более структурированная версия Lean нежели Kanban, в которую добавлено больше планирования для экономии ресурсов, повышения качества, также снижения количества брака и проблем.



Конечная цель проекта – удовлетворение заказчика качеством продукта, которого можно добиться при помощи непрерывного процесса улучшения всех аспектов проекта, основанном на тщательном анализе показателей. В концепции 6 сигма уделяется отдельное внимание устранению возникающий проблем.

Для этого было предложен процесс из 5 шагов, известных как DMEDI:

* **Определение (Define):** Первый этап очень похож на ранние этапы других систем проектного управления. На нём определяется содержание проекта, собирается информация о предпосылках проекта, ставятся цели.
* **Измерение (Measure):** 6 сигм ориентирована на сбор и анализ количественных данных о проекте. На данном этапе происходят определяется, какие показатели будут определять успех проекта и какие данные нужно собирать и анализировать.
* **Исследование (Explore):** На стадии исследования менеджер проекта решает, каким же образом команда может достичь поставленных целей и исполнить все требования в срок и в рамках бюджета. На данном этапе очень важно нестандартное мышление руководителя проектов при решении возникших проблем.
* **Разработка (Develop):** На данном этапе реализуются планы и решения, принятые на предыдущих этапах. Важно понимать, что на данном этапе необходим детальный план, в котором описаны все действия, необходимые для достижения поставленных целей. Также на данном этапе измеряется прогресс проекта.
* **Контроль (Control):** Ключевой этап в методологии 6 сигм. Его основная задача – долгосрочное улучшение процессов реализации проектов. Данный этап требует тщательного документирования извлечённых уроков, анализа собранных данных и применения полученных знаний как в проектах, так во всей компании в целом.

6 сигм очень похожа на Kanban, только с установленными этапами реализации задач – планированием, определением целей и тестированием качества. Вероятнее всего, встреч команды при применении 6 сигм будет значительно больше, чем при Kanban, но зато процесс реализации проектов более структурирован и команде сложнее сбиться с пути. И, как и Kanban, 6 сигм можно относительно легко адаптировать к нуждам конкретной компании или команды. Жёстким требованием является лишь тщательное измерение и контроль показателей проекта на этапах реализации – без этого невозможно постоянное долгосрочное улучшение процессов реализации проекта.

**Сильные стороны 6 сигм**

Концепция 6 сигм предоставляет чёткую схему для реализации проектов и постоянного улучшения процессов. Определяя цели, затем тщательно анализируя их и пересматривая вы получаете количественные данные для более глубокого понимания проекта и принятия более качественных решений. И хотя сбор, анализ данных и извлечение уроков могут занять определённое время, это позволит улучшить и оптимизировать процессы реализации проекта и сэкономить таким образом ресурсы в будущем.

6 сигм подходит для трудных проектов, в которых много новых и сложных операций. Данный подход позволяет реализовывать элементы проекта, учиться на ошибках и повышать качество в будущем.

**Слабые стороны 6 сигм**

Проблема 6 сигм в том, пусть основной декларируемой целью является снижение затрат и повышение эффективности, но удовлетворение Заказчика часто вырывается на первый план. Учитывая некоторые различия в целях на разных этапах проекта, часто у команд возникает путаница в приоритетах, и избежать этого не просто.

Кроме того, основной лейтмотив 6 сигм: «Всё всегда можно сделать ещё лучше». Это может демотивировать сотрудников, не чувствующих удовлетворения от проделанной работы. Кроме того, если проект единичный и компания не планирует в будущем реализовывать подобные проекты, все затраты на анализ и извлечение уроков могут оказаться напрасными.

PRINCE2

НАСА – не единственная государственная организация, которая внесла вклад в развитие проектного управления. Британское Правительство давно оценило эффективность проектного управления, и в 1989 году была создана британская методология PRINCE2. Название произошло от акронима «**PR**ojects **IN** **C**ontrolled **E**nvironments version **2**», что переводится как «Проекты в контролируемой среде версия 2». В отличие от гибких методов, PRINCE2 не использует итеративный подход к проекту. Если сравнивать PRINCE2 другими продуктами, то его можно сравнить с гибридом классического подхода к проектному управлению и концентрации на качестве из 6 сигм.



Методология PRINCE2 в отличие от, например, свода знаний PMBOK не содержит:

* Специализированных аспектов управления проектом, например, отраслевых;
* Конкретных практик и инструментов управления проектами, таких как диаграмма Гантта, WBS и т.п.

PRINCE2 концентрируется на управленческих сторонах проекта, выраженных в 7 принципах, 7 процессах и 7 темах проекта.

* 7 принципов определяют общие правила управления проектами по PRINCE2, определяют базу методологии;
* 7 процессов определяют шаги продвижения по проектному циклу;
* 7 тем – аспекты, по которым проводится контроль для достижения успеха проекта.

Кроме того, PRINCE2 рекомендует адаптировать методологию под каждую конкретную организацию.

В начале проекта PRINCE2 предлагает нам определить 3 основных аспекта проекта:

* Бизнес-аспект (Принесёт ли этот проект выгоду?)
* Потребительский аспект (Какой нужен продукт, что мы будем делать?)
* Ресурсный аспект (Достаточно ли у нас всего, чтобы достичь цели?)

В PRINCE2 более чётко определённая структура команды проекта, чем у большинства подходов к проектному управлению. Это связано с тем, что PRINCE2 ориентирован на масштабные государственные проекты и крупные организации.



Согласно PRINCE2 у каждого члена команды есть своя чёткая роль в каждом из 7 процессов:

* **Начало проекта (Starting** **up a** **project):**В ходе данного процесса назначается менеджер проекта и определяются общие требования к характеристикам продукта. Менеджер проекта, чья основная задача – внимание к деталям, отчитывается перед Управляющим комитетом проекта, который отвечает за общее руководство проектом. Именно Управляющий комитет следит за тем, чтобы проект не сбился с курса, и он же полностью отвечает за успех проекта.
* **Инициация проекта (Initiation a** **project):**В ходе данного процесса менеджер проекта составляет «Документацию по инициации проекта», в которой содержится план проекта по стадиям. Стадии могут длиться разное количество времени, но, как и в классическом подходе, они следуют строго друг за другом.
* **Руководство проектом (Directing** **a** **project):**Данный процесс предоставляет возможность Управляющему комитету нести общую ответственность за успех проекта, не погружаясь в детали, которые находятся в границах полномочий менеджера проекта.
* **Контроль стадии (Controlling** **a** **stage):**При реализации проекта, даже в идеальных условиях, будут вноситься определённые изменения. Процесс «Контроль стадии» реализует один из принципов PRINCE2 – принцип управления по исключениям. В обязанности менеджера проекта входит отслеживать в ходе выполнения стадии отклонения от плановых параметров проекта по срокам, содержанию, бюджету и др. Если эти отклонения превышают данные руководителю проекта Управляющим комитетом полномочия (в терминологии PRINCE2 – допуски), менеджер проекта обязан проинформировать Управляющий комитет и предложить пути выхода из ситуации.
* **Управление созданием продукта (Managing** **Product** **Delivery):**Процесс управления созданием продукта представляет собой взаимодействие менеджера проекта и менеджера команды по созданию одного из продуктов проекта. В обязанности менеджера проекта в данном процессе входит делегирование полномочий по созданию продукта менеджеру команды и приемка созданного продукта.
* **Управление границами стадии (Managing** **a** **stage** **boundary):**В ходе данного процесса менеджер проекта предоставляет Управляющему комитету всю необходимую информацию для оценки результатов пройденной стадии и принятия решения о переходе на следующую стадию.
* **Завершение проекта (Closing a** **project):**Одно из отличий PRINCE2 в том, что процесс завершения проекта не выделяется в отдельный этап или стадию, как в классическом подходе, а выполняется в рамках финальной стадии создания продукта. Цель процесса – подтвердить, что продукт проекта принят, или проект больше не может принести ничего полезного.

PRINCE2 может быть адаптирован для проектов любого масштаба и любой предметной области. Методология предлагает конкретные рекомендации по изменению жизненного цикла проекта, ролевой модели и набора обязательных документов в соответствии с потребностями проекта.

**Сильные стороны PRINCE2**

* Адаптируемость к особенностям организации;
* Наличие чёткого описания ролей и распределения ответственности;
* Акцент на продуктах проекта;
* Определённые уровни управления;
* Фокус на экономической целесообразности;
* Последовательность проектной работы;
* Акцент на фиксации опыта и постоянном совершенствовании.

**Слабые стороны PRINCE2**

* Отсутствие отраслевых практик;
* Отсутствие конкретных инструментов для работы в проекте.