

Программная инженерия

Управление программной инженерией (Software Engineering Management)

Глава базируется на IEEE Guide to the Software Engineering Body of Knowledge⁽¹⁾ - SWEBOK[®], 2004. Содержит перевод описания области знаний SWEBOK[®] "Software Engineering Management", с комментариями и замечаниями⁽²⁾.

Сергей Орлик.

⁽¹⁾ Copyright © 2004 by The Institute of Electrical and Electronics Engineers, Inc. All rights reserved.

⁽²⁾ расширения SWEBOK отмечены, следуя IEEE SWEBOK Copyright and Reprint Permissions: This document may be copied, in whole or in part, in any form or by any means, as is, or with alterations, provided that (1) alterations are clearly marked as alterations and (2) this copyright notice is included unmodified in any copy.

Программная инженерия

Управление программной инженерией (Software Engineering Management)

Программная инженерия	2
Управление программной инженерией (Software Engineering Management)	2
1. Инициирование и определение содержания (Initiation and Scope Definition)	6
1.1 Определение и обсуждение требований (Determination and Negotiation of Requirements)	7
1.2 Анализ осуществимости. Технические, операционные, финансовые, социальные/политические аспекты. (Feasibility Analysis. Technical, Operational, Financial, Social/Political.)	7
1.3 Процесс оценки и пересмотра требований (Process of Review and Revision of Requirements)	7
2. Планирование программного проекта (Software Project Planning)	7
2.1 Планирование процесса (Process Planning)	8
2.2 Определение результатов (Determine Deliverables)	8
2.3 Оценка усилий, расписания и стоимостных ожиданий (Efforts, Schedule and Cost Estimation)	8
2.4 Распределение ресурсов (Resource Allocation)	9
2.5 Управление рисками (Risk Management)	9
2.6 Управление качеством (Quality Management)	10
2.7 Управление планом проекта (Plan Management)	10
3. Выполнение программного проекта (Software Project Enactment)	10
3.1 Реализация планов* (Implementation of Plans)	10
3.2 Управление контрактами с поставщиками (Supplier Contract Management)	10
3.3 Реализация процесса по ведению измерений (Implementation of Measurement Process)	11
3.4 Процесс мониторинга (Monitor Process)	11
3.5 Процесс контроля (Control Process)	11
3.6 Ведение отчетности (Reporting)	11
4. Обзор и оценка (Review and Evaluation)	12
4.1 Определение удовлетворения требованиям (Determining Satisfaction of Requirements)	12
4.2 Оценка продуктивности/результативности (Reviewing and Evaluation Performance)	12
5. Закрытие (Closure)	12
5.1 Определение <критериев> закрытия проекта (Determining Closure)	12
5.2 Работы по закрытию проекта (Closure Activities)	13
6. Измерения в программной инженерии (Software Engineering Measurement)	13
6.1 Установление и поддержка процесса ведения измерений (Establish and Sustain Measurement Commitment)	13
6.2 Планирование процесса измерений (Plan the Measurement Process)	14
6.3 Выполнение процесса измерений (Perform the Measurement Process)	15
6.4 Оценка измерений (Evaluate Measurement)	15

Управление программной инженерией может быть определено как приложение вопросов управления (management activities) – планирования, координации, количественной оценки, мониторинга, контроля и отчетности – к инженерной деятельности для систематического, упорядоченного и количественно измеряемого обеспечения разработки и сопровождения программных систем (IEEE 610.12-90, Standard Glossary for Software Engineering Terminology).

Таким образом, область знаний “Управление программной инженерией” определяет аспекты управления и количественной оценки в программной инженерии. Измерения являются важным аспектом для всех областей знаний SWEBOK и соответствующая тема также включена и в описании данной области знаний.

В принципе, корректно утверждать, что возможно управлять программной инженерией так же, как и любым другим комплексным процессом. В то же время, существуют аспекты, специфичные для программных продуктов, а процессы жизненного цикла программных систем, в какой-то мере, усложняют достижение необходимого уровня эффективности управления. Среди таких усложняющих факторов:

- Восприятие клиентов (потребителей, заказчиков) таково, что часто отсутствует с их стороны понимание сложности, порожденной самой природой программной инженерии, в частности связанной с влиянием изменяющихся требований.
- Практически неизбежно изменение или появление новых клиентских требований, в следствие функционирования процессов программной инженерии.
- В результате, программные системы часто строятся с применением итеративных процессов, вместо последовательного выполнения и закрытия работ (задач).
- Уровень новизны и сложности программных систем часто крайне высок (и не позволяет в полной мере применять уже существующие наработки и опыт, *прим. автора*).
- Применяемые технологии обладают высокой скоростью изменения, обновления и устаревания

Что касается программной инженерии, управленческая деятельность в этой области происходит на трех уровнях:

- Организационное управление и управление инфраструктурой
- Управление проектами
- Планирование и контроль программ количественной оценки

Последние два уровня описаны в данной области знаний, что никак не принижает значимости общих вопросов управления организационными аспектами и инфраструктурой.

Хотя все области знаний тесно связаны с другими дисциплинами, связь данной области знаний с общими вопросами менеджмента особенно важны, что и будет более подробно, в отличие от других областей знаний, представлено в ниже. Вопросы организационного менеджмента важны с точки зрения влияния на программную инженерию, например, в контексте управления политиками/полномочиями сотрудников или других внутрикорпоративных стандартов, в рамках которых выполняется любая деятельность (например, с точки зрения отчетности по занятости сотрудников), в том числе - инженерная. Такие политики подвержены влиянию со стороны требований к организации *эффективного процесса разработки и сопровождения* и, на практике, бывает необходимо адаптировать общие и создать специальные для инженерной деятельности внутренние организационные стандарты, обеспечивающие *эффективное управление программной инженерией*. Эти политики являются основой для решения долгосрочных задач улучшения процессов и повышения производительности труда специалистов, вовлеченных с работы по созданию сопровождению программного обеспечения.

Другим важным аспектом управления является управление персоналом через политики и процедуры найма и приема на работу, обучения, и мотивации специалистов, помощи в развитии навыков для дальнейшего карьерного роста (*mentoring in career development*). Все это требует внимания не только в контексте проекта, но в рамках всей организации. Для инженеров-программистов особо важными, в частности, являются вопросы обучения и индивидуального внимания менеджмента. В большой степени это связано с постоянно развивающимися технологиями и потребностью в обновлении и расширении знаний для эффективного решения поставленных задач. Часто не придают необходимого внимания вопросам коммуникаций между сотрудниками. На самом деле управление коммуникациями, создание естественных условий (в agile-практиках им придается особое внимание) для их развития – один из ключевых элементов повышения не только продуктивности команд разработки и сопровождения, но и, например, точности получаемых от пользователей требований и запросов на изменения, то есть любой информации, которая передается между людьми и значима для успешного решения поставленных задач. Наконец, управление портфелями (проектов разработки и работ по сопровождению) позволяет сформировать и развивать общее видение в отношении всех существующих, обновляемых и создаваемых программных активов на уровне ИТ-подразделения и в организации, в целом. Все это, в конечном счете, обеспечивает и более эффективное управление ресурсами, а, значит, и возможность интенсивного, а не экстенсивного развития организации, в которой инновации начинают играть одну из ключевых ролей.

Вместе с осознанием специфики управленческой деятельности в приложении к программной инженерии, ИТ-специалистам необходимо понимать и ключевые аспекты общего менеджмента и управления проектами.

Организационная культура, нормы поведения, аспекты корпоративного управления в вопросах приобретения и поставки, управления цепочками поставок (supply chain management), маркетинг, продажи, партнерства и т.п. – все это влияет, хотя и неявно, на организационные процессы программной инженерии.

В отношении данной области знаний особо уместно подчеркнуть значимость вопросов управления проектами (project management), так как *“конструирование имеющих ценность программных артефактов”* (к которым относятся требования, модели, документация, тесты и т.п., *прим. автора*) *обычно ведется в форме проектов или программ проектов*. Принимая это во внимание, создатели SWEBOOK особо отмечают связь данной области знаний с обсуждавшимся уже в этой книге Руководством к Своду Знаний по Управлению Проектами *PMBOK* (A Guide to the Project Management Body of Knowledge. PMBOK® Guide). В контексте управления программной инженерией следует понимать важность соответствующих областей знаний PMBOK:

- Управление интеграцией проекта (project integration management)
- Управление содержанием проекта (project scope management)
- Управление сроками проекта (project time management)
- Управление стоимостью проекта (project cost management)
- Управление качеством проекта (project quality management)
- Управление человеческими ресурсами проекта (project human resource management)
- Управление коммуникациями проекта (project communication management)

Наравне с ними, с точки зрения автора, необходимо уделять не меньшее внимание и другим областям знаний управления проектами:

- Управление рисками проекта (project risk management)
- Управление поставками проекта (project procurement management)

SWEBOOK отмечает, что, несомненно, области знаний управления проектами имеют непосредственное влияние на решение вопросов управления инженерной деятельностью в области программного обеспечения. Не имеет смысла, да и просто невозможно дублировать в SWEBOOK содержание PMBOK. Вместо этого, PMBOK рассматривается как ключевой источник информации и знаний по управлению проектами, настоятельно рекомендуемый всем лицам, в той или иной степени вовлеченных в управленческую деятельность в программных проектах. Таким образом, естественно, что управление проектами можно найти в области знаний SWEBOOK *“Связанные дисциплины”* (Related Disciplines).

Данная область знаний состоит из пяти секций, посвященных процессам управления программной инженерией и еще одной секции, касающейся вопросов измерений и количественных оценок в управлении. Хотя эти два аспекта (управление и измерения) часто рассматриваются отдельно и, в самом деле, обладают многими уникальными аспектами, их тесная взаимосвязь играет важную роль в этой области знаний. К сожалению, сложилось, во многих случаях, обоснованное [Chaos, 2004] восприятие индустрии программного обеспечения как недостаточно зрелой, в силу частых срывов сроков, превышения бюджетных ограничений, недостаточного качества продуктов, неопределенной функциональности и других причин. Управление, ориентированное на измерения, как один из основных принципов любой инженерной деятельности, может серьезно помочь в изменении сложившейся неблагоприятной ситуации и формировании положительного восприятия программной индустрии потребителями (пользователями и заказчиками). По-сути, управление без измерений, количественных или качественных, приводит к отсутствию прогресса в достижении целей, а измерения без управления – к потере контекста и целей. Однако, в то же время, управление и измерения без необходимого и достаточного уровня знаний становится неэффективным и, часто, превращается в самоцель (что приводит, по мнению автора к излишней бюрократизации и неадекватной загруженности ресурсов). Таким образом, управленческая деятельность, в общем плане (включая количественные и качественные оценки), должна проводиться сбалансировано с другими аспектами программной инженерии, не превращая Software Engineering Management (SEM) в дорогостоящую, но бесполезную работу. Эффективный менеджмент требует комбинации соответствующих систематических и упорядоченных подходов в управлении и соответствующего опыта*.

* например, на практике просто невозможно добиться статуса PMP – Project Management Professional по версии Project Management Institute (PMI), если претендент не обладает серьезным практическим опытом управления

проектами и пытается сдать соответствующий экзамен только на основе штудирования PMBOK и теоретических “изысканий”.

Прежде, чем детализировать данную область знаний, необходимо дать рабочие определения для понятий “процесс управления” и “измерения”:

- *Процесс управления (Management process)* описывает действия (работы - activities), предпринимаемые для обеспечения того, что процессы программной инженерии выполняются в согласовании с политиками, целями и стандартами, принятыми в организации
- *Измерения (Measurement)* связаны с определением величин и характеристикой различных аспектов программной инженерии (продуктов, процессов и т.п.), а также разработкой на их основе моделей* с использованием статистических методов (и данных), экспертных знаний и других техник.

* В данном контексте, по мнению автора, SWEBOOK подразумевает, что полученные модели используются для идентификации и анализа рисков, планирования и совершенствования процессов программной инженерии (процессов жизненного цикла, включая процессы сопровождения), а также процесса управления.

Секции (подобласти) данной области знаний, связанные с управлением программной инженерией, тесно связаны с секцией измерений (количественной оценки).

Вполне естественно, что данная область знаний тесно связана с другими областями знаний SWEBOOK и ее необходимо рассматривать в контексте других областей знаний. Стоит особо отметить следующие аспекты применения других областей знаний в управлении программной инженерией и, особо, в управлении программными проектами:

- *Требования к программному обеспечению (Software Requirements)* – соответствующие действия по работе с требованиями (в первую очередь, их определение) указанной области знаний должны выполняться в фазе инициирования (Initiation) и определения содержания (Scope Definition) проектов.
- *Конфигурационное управление (Software Configuration Management)* – связано с идентификацией, контролем, учетом статуса (активов проекта, включая запросы на изменения, *прим. автора*) и аудитом конфигураций (в терминах конфигурационного управления) в сочетании с управлением релизами (release management) и развертыванием программных систем.
- *Процесс программной инженерии (Software Engineering Process)* – процессы и продукты тесно связаны; указанная область знаний включает аспекты измерений продуктов и процессов.
- *Качество (Software Quality)* – качество является одной из постоянных целей управления и большого комплекса соответствующих работ, которыми необходимо управлять.

Данная область знаний рассматривает управление программной инженерией в терминах организационного процесса, включающего управление процессами и проектами. Структурная декомпозиция этой области знаний основывается и на определении соответствующих тем и на рассмотрении жизненного цикла. При этом, основной отправной точкой дальнейшей детализации является процесс управления программными проектами (в оригинале SWEBOOK в данной области знаний активно используется термин software engineering project, *прим. автора*). Таким образом, структура декомпозиции управления программной инженерией включает шесть основных секций (подобластей), из которых первые пять, в основном, следуют стандарту IEEE (ISO/IEC, ГОСТ) 12207 в части “Процесса управления” (Management Process). Вот эти шесть секций данной области знаний:

- *Инициирование и определение содержания (Initiation and scope definition)* – касается принятия решения о начале программного проекта
- *Планирование программного проекта (Software project planning)* – относится к работам, предпринимаемым для подготовки к успешному ведению программно-инженерной деятельности с точки зрения управления
- *Выполнение программного проекта (Software project enactment)* – касается общепринятых (general accepted) действий по управлению программной инженерией в процессе проведения соответствующих инженерных работ
- *Обзор и оценка (Review and evaluation)* – относится к работам по проверке того, что получаемый программный продукт отвечает заданным целям, требованиям, ограничениям и т.п.

- *Закрытие <проекта> (Closure)* – относится к фиксированию результатов программного проекта после передачи полученного программного продукта в эксплуатацию.
- *Измерения в программной инженерии (Software engineering measurement)* – касается разработки и реализации программ по измерению (ведению количественной оценки) в организациях (в общем смысле, т.е. группах, подразделениях, компаниях и т.п.), занимающихся инженерной деятельностью в области программного обеспечения.

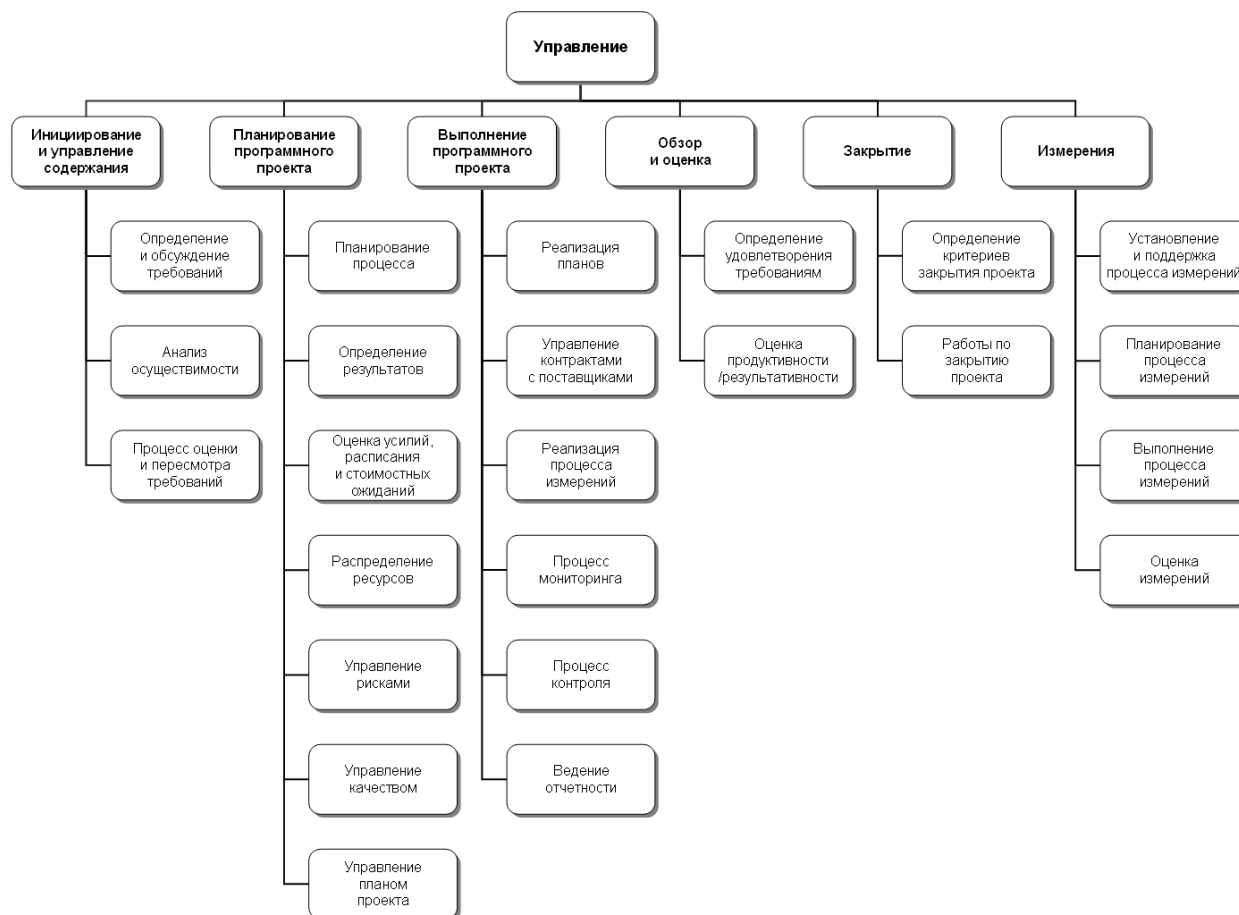


Рисунок 1. Область знаний “Управление программной инженерией” [SWEBOK, 2004, с.8-2, рис. 1]

Здесь автор не может не сделать замечание, связанное со структурированием этой области знаний. Обсуждаемая область знаний, действительно, тесно связана с дисциплиной управления проектами. Более того, речь идет о приложении управления проектами к программной инженерии. В этом контексте кажется уместным сопоставить предлагаемый в SWEBOK цикл “*Initiation & scope definition – Planning – Enactment – Review & evaluation – Closure*” с процессными группами PMBOK “*Initiation – Planning – Execution – Monitoring & Controlling – Closing*”. Как мы видим, в PMBOK роль работ SWEBOK “Обзор и оценка” (Review and evaluation) играют действия по мониторингу и контролю процессов - “Monitoring & Controlling Processes”. Таким образом, с учетом реального содержания секции “Software project enactment”, ее название и переведено автором как “Выполнение программного проекта”, хотя “enactment” в большей степени подразумевает формальный “запуск” работ. Конечно, перевод мог звучать и как “исполнение” (например, следуя PMBOK), однако, такой перевод, по мнению автора, все же несет слишком формальный оттенок, что, субъективно, не соответствует ряду методологических подходов (в первую очередь agile) и культурных “ограничений”.

1. Инициирование и определение содержания (Initiation and Scope Definition)

Данная секция фокусируется на наборе действий, связанных с эффективным определением требований к программному обеспечению с использованием различных методов извлечения требований, а также оценкой осуществимости проекта с различных точек зрения. Если проект

признан осуществимым, следующей задачей является специфицирование процедур проверки и изменения требований (см. область знаний “Требования к программному обеспечению” – Software Requirements).

1.1 Определение и обсуждение требований (Determination and Negotiation of Requirements) – выбор и применение методов определения (извлечения), анализа (например, моделирования сценариев use case), специфицирования и проверки (например, прототипирования) требований, принимая в внимание позицию различных заинтересованных лиц. Это является первичными работами, необходимыми для определения содержания, целей и ограничений проекта. Данные работы важны всегда, так как позволяют определить четкие границы задач, необходимых для выполнения, в частности, это особенно заметно для проектов, обладающих большой степенью “новизны” (идет ли речь о технологических аспектах проекта, его масштабах, методах и т.п., *прим. автора*).

1.2 Анализ осуществимости. Технические, операционные, финансовые, социальные/политические аспекты. (Feasibility Analysis. Technical, Operational, Financial, Social/Political.)

Инженеры должны убедиться в том, что для успешного завершения проекта (в заданные сроки, в рамках бюджета и т.п.) доступны необходимые возможности (capabilities) и ресурсы, будь то люди (те или иные специалисты), экспертиза (опыт, знания, навыки), средства (например, инструментарий), инфраструктура и поддержка (как внутренняя, так и внешняя, например, со стороны старших менеджеров организационной структуры, отвечающей за разработку, и ключевых менеджеров или других заинтересованных лиц со стороны “заказчика”). Это часто требует хотя бы приблизительной оценки усилий и стоимости с использованием соответствующих методов (например, техники, в которой экспертная оценка базируется на аналогиях).

1.3 Процесс оценки и пересмотра требований (Process of Review and Revision of Requirements)

Учитывая неизбежность изменений, жизненно важно определить и согласовать с заинтересованными лицами процедуры (например, в контексте деятельности по управлению изменениями) в рамках которых будут проводиться оценка и пересмотр требований. Это однозначно предполагает, что требования не будут неизменны, но могут и должны корректироваться в соответствии с заданными и детализированными процессами (например, design review – оценка дизайна). Если изменения приняты, необходимо проводить анализ зависимостей (traceability analysis) и рисков (см. далее тему 2.5 “Управление рисками” – Risk Management) для оценки влияния рассматриваемых изменений. С точки зрения автора, такой анализ необходимо проводить, в общем случае, и при рассмотрении “изменения” для принятия решения о передаче его “в работу” или отклонении (например, в силу обоснованных причин, таких как проектные решения, позволяющих отметить его как реализуемое в следующей версии), и уже более детально, если изменение требования(ий) решено реализовать (в силу, например, контрактных обязательств со стороны исполнителя) и необходимо определить, как именно такое изменение повлияет на существующую систему и какие работы необходимо выполнить, чтобы удовлетворить обновленному(ым) требованиям. SWEBOK отмечает, что управление изменениями полезно и при оценке результатов проекта (программного продукта, программной системы), так как содержание и требования формируют основу оценки успешности проекта. (см. также секцию “Контроль конфигураций” области знаний Software Configuration Management).

2. Планирование программного проекта (Software Project Planning)

Процесс планирования является итеративным* и базируется на содержании, требованиях и оценке осуществимости проекта. С точки зрения автора, здесь стоит напомнить, что различные фазы проекта перекрываются (что, например, специально отмечает PMBOK) и вместе с определением содержания, детализацией требований и проведением анализа осуществимости, параллельно с этим сам разрабатываемый план проекта в той или иной степени детализируется и формируется определенный комплекс работ, оцениваются необходимые ресурсы и временные границы работ, и т.п. SWEBOK, основываясь на такой позиции, говорит, что при планировании также оцениваются и отбираются соответствующие процессы жизненного цикла. Там где это уместно, проект детализируется в виде структурной декомпозиции работ, для которых отмечены ассоциируемых с их завершением результаты и их характеристики. Такие характеристики, обычно, связаны с вопросами качества и другими установленными атрибутами требований, соблюдением сроков выполнения работ, усилиями, стоимостью и т.д. Ресурсы распределяются по задачам таким образом, чтобы

обеспечить оптимальную продуктивность (на персональном, командном и организационном уровне), использование средств (инфраструктуры, инструментов,...) и оборудования, а также строгое соблюдение проектного расписания. Также должно проводиться управление рисками и, в частности, необходимо определить “профиль рисков”, принятый соответствующими заинтересованными лицами. Как составная часть планирования, необходимо определить необходимые процессы обеспечения качества в форме соответствующих процедур и обязанностей (responsibilities) по оценке, проверке, аттестации и аудиту качества (см. область знаний “Качество программного обеспечения” – Software Quality). Безусловно, что должны быть определены процессы и обязанности в части управления планом проекта, его оценкой и порядка пересмотра различных аспектов проекта.

* позднее, при обсуждении жизненного цикла и, в частности, спиральной модели разработки, мы будем рассматривать ключевые идеи итеративного подхода, *прим. автора*.

2.1 Планирование процесса (Process Planning)

С учетом содержания и требований конкретного проекта необходимо выбрать, адаптировать и использовать соответствующую модель процессов жизненного цикла (например, спиральную, с эволюционным прототипированием). Также должны быть выбраны методы и инструменты. На уровне проекта, методы и инструменты используются для декомпозиции проекта в виде набора задач (tasks), с ассоциированными входами, выходами и условиями завершения (completion), например, в форме структуры декомпозиции работ – WBS (work breakdown structure). Это влияет на высокоуровневое (первичное, *прим. автора*) определение проектного расписания и организационной структуры <проектной команды>.

2.2 Определение результатов (Determine Deliverables)

Должен быть определен результат выполнения каждой задачи (например, описание архитектуры, отчет по анализу, набор тестов и т.п.), то есть какие активы/артефакты мы должны получить по выполнению соответствующей задачи проекта. При этом оцениваются возможности повторного использования программных компонент, созданных ранее, в процессе разработки других программных продуктов, и потенциал применения готовых к использованию компонент из 3-их источников. Должно быть определено, какие именно компоненты будут использоваться и как они будут получены (через каких поставщиков).

Такие компоненты, обычно, называют “off-the-shelf”, подразумевая в настоящее время не только коммерческое, но и общедоступное программное обеспечение, если его использование обосновано в рамках данного проекта. По глубокому убеждению автора, *анализ и выбор соответствующих компонент может и, в подавляющем большинстве случаев, должен рассматриваться как самостоятельная задача процесса планирования* в контексте сформулированных высокоуровневых требований, определенного содержания и базовых ограничений проекта, таких как сроки, ресурсы, стоимость). Это позволяет четко разделить, в том числе, в контексте затрат, что именно будет разрабатываться самостоятельно (может быть как отдельный “суб-проект”), что будет использоваться из 3-их источников (3rd party), а что будет являться содержательным (с точки зрения проекта) результатом работ, то есть его непосредственным активом, обладающим, например, функциональной, для данного проекта, нагрузкой.

2.3 Оценка усилий, расписания и стоимостных ожиданий (Efforts, Schedule and Cost Estimation)

Ожидаемые пределы усилий, необходимых для решения каждой задачи (task) проекта основываются на разбиении задач, их входах и выходах. Для этого используется калиброванная (calibrated - настроенная для заданных условий) модель ожиданий (estimation model), базирующаяся на исторических данных по усилиям, связанным с объемом задачи (size-effort historical data, часто определяется как человеко-месяцы к функциональным точкам или количеству строк кода, *прим. автора*). Также, для оценки усилий могут применяться и другие методы, например, экспертная оценка или оценка по типу приложения (встроенное, телекоммуникационное), квалификации проектной группы и т.п.

Кроме того, необходимо идентифицировать связи и зависимости между задачами (tasks dependencies) и потенциально критические аспекты (bottlenecks) проекта. Такие работы могут быть проведены с использованием, например, метода анализ критического пути (critical path analysis –

достаточно распространенный метод, относящийся к общей дисциплине управления проектами, применимый и для проектов программных систем, *прим. автора*). Если возможно, критические аспекты должны быть разрешены, а для задач определены ожидаемые сроки выполнения (расписание), включающие начало, длительность и окончание (например, в форме PERT-диаграмм*).

* PERT анализ (Program, Evaluation, and Review Technique) – техника оценки ожиданий в отношении длительности (duration) задач проекта, проводимая на основе определения среднего весового значения трех оценок длительности – пессимистической, оптимистической и ожидаемой (то есть наиболее вероятной, при первичной оценке). Аналогичная техника может и часто используется для оценки усилий (effort), необходимых для реализации задачи. Наибольший эффект дает сочетание различных методов оценки. В то же самое время, чем больше методов оценки используется, тем более трудоемкой (а, следовательно, и ресурсоемкой) становится такая работа, поэтому задача менеджмента – определить наиболее оптимальный и эффективный для данного проекта набор методов и техник, используемых в процессе планирования и корректировки.

Требования к ресурсам (люди, инструменты) транслируются в стоимостные ожидания.

В совокупности, вся эта деятельность является итеративной и должна обсуждаться и проводиться до тех пор, пока не будет достигнут консенсус между соответствующими заинтересованными лицами – в первую очередь, менеджментом <проекта> и инженерами <входящими в команду проекта>.

2.4 Распределение ресурсов (Resource Allocation)

С задачами (для которых назначены сроки), должны быть ассоциированы оборудование, средства и, конечно, люди. Это подразумевает распределение (назначение или принятие, в зависимости от стиля и формы управления) обязанностей/ответственности. Для этого может, например, использоваться диаграмма Ганта (Gantt chart). Эта деятельность определяется и ограничивается доступностью ресурсов, их оптимальным использованием в заданном контексте и вопросами, связанными с персоналом (например, продуктивностью конкретных лиц и группы, в целом, организационной и командной структурой, подразумевая специфику коллектива, наравне со штатным расписанием и другие вопросы).

По мнению автора, крайне необходимо выделить как самостоятельную тему данной секции *вопросы управления персоналом – people management*, уделив особое внимание аспектам экспертизы (не стоит путать с ролями/обязанностями) и лидерства специалистов в проектной команде. К этой же теме, также стоит отнести *вопросы обучения*, прохождения тренингов специалистами проектной команды. Наконец, к теме ресурсов имеет непосредственное отношение и *определение необходимости и объема привлечения внешних консультантов* (не являющихся сотрудниками ни исполнителя, ни заказчика), к сожалению, не упоминаемое здесь в SWEBOOK, но крайне важное, по опыту автора, для успешности проекта, обладающего высокой степенью новизны (например, в терминах используемых технологий и, особенно, применения тех или иных архитектурных решений).

2.5 Управление рисками (Risk Management)

В части управления рисками должны проводиться:

- идентификация и анализ рисков – что, когда и почему может быть сделано неверно и к чему это может привести;
- оценка критических рисков – какие из рисков наиболее значительны (если им не уделять должного внимания) и что необходимо сделать, чтобы их избежать;
- смягчение рисков (risk mitigation) и планируемость непредвиденных обстоятельств (contingency planning) – формирование стратегии, касающейся рисков и управление профилями рисков.

Для идентификации и оценки рисков необходимо применять соответствующие методы и техники (например, построение дерева решений – decision tree или моделирование процессов – process simulation). Кроме того, со всеми заинтересованными лицами необходимо определить правила и политики прекращения проекта.

Наравне с идеями общего управления рисками, важно понимать и управлять рисками, уникальными для деятельности в области программной инженерии, например, тенденция добавлять в получаемый программный продукт функциональные и другие возможности, неопределенные на уровне

требований или риски, заложенные в самой природе программного обеспечения, связанные, в первую очередь с его сложностью и архитектурно-технологической новизной, присутствующей, в той или иной степени, в любом программном проекте.

2.6 Управление качеством (*Quality Management*)

Качество определяется в терминах атрибутов, значимых для данного конкретного проекта и/или ассоциированного с ним продукта. Атрибуты могут выражаться как качественно, так и количественно. Эти характеристики качества определяются в спецификации требований к программному обеспечению (см. область знаний “Требования к программному обеспечению” – Software Requirements).

Отправной точкой для соблюдения качества является набор индикаторов, соответствующих ожиданиям заинтересованных лиц. На этой стадии (как мы помним, речь идет о планировании проекта) также специфицируются процедуры, связанные с проведением SQA-деятельности (деятельности по обеспечению качества – software quality assurance) на протяжении всех процессов жизненного цикла и для проверки и аттестации (V&V – verification and validation) для получаемого продукта и всех активов (артефактов) проекта (см. область знаний “Качество программного обеспечения” – Software Quality).

2.7 Управление планом проекта (*Plan Management*)

Наравне с другими аспектами ведения проекта, должно быть определено как проект будет управляться и как будет управляться план проекта. Ответственность, мониторинг и контроль проекта должны соответствовать выбранному процессу программной инженерии и сущности проекта, отражая также в виде различных артефактов именно то, что будет использоваться в процессе управления. При этом, в изменяющемся окружении принципиально важно, чтобы и сам план проекта был управляем. Это требует строгого соблюдения планов, которые должны быть систематически направляемы, контролируемы, оцениваемы, по которым будет вестись отчетность и, там где это применимо, корректируемы. Планы, ассоциированные с другими процессами поддержки, ориентированными на управление, также должны быть управляемы соответствующим образом (например, это касается вопросов документирования, конфигурационного управления и разрешения проблем).

3. Выполнение программного проекта (*Software Project Enactment*)

План проекта реализуется за счет выполнения процессов, представленных в плане. Следование плану на протяжении выполнения проекта связано с ожиданиями, что соблюдение <корректно составленного> плана приводит к успешному удовлетворению требований заинтересованных лиц и достижению целей проекта. Основой для успешного выполнения проекта является управленческая деятельность по ведению оценки и измерений, мониторинга, контроля и отчетности.

3.1 Реализация планов* (*Implementation of Plans*)

Проект инициируется и проектные работы выполняются в соответствии с планом. В процессе выполнения используются соответствующие ресурсы (например, усилия персонала, бюджет) и производятся необходимые результаты (deliverables; активы, артефакты проекта – например, архитектурные документы, тестовые сценарии).

* в SWEBOOK используется как “план проекта” в единственном числе, так и “планы” – во множественном числе, подразумевающие, судя по всему, отдельные задачи проекта.

3.2 Управление контрактами с поставщиками (*Supplier Contract Management*)

Включает подготовку и выполнение соглашений с поставщиками, мониторинг деятельности поставщиков, принятие у поставщиков продуктов, использование и интеграцию этих продуктов в рамках проектных работ.

3.3 Реализация процесса по ведению измерений (Implementation of Measurement Process)

Данный процесс выполняется на протяжении всего проекта, обеспечивая сбор всех необходимых данных (см. 6.2 Plan the Measurement Process и 6.3 Perform the Measurement Process).

3.4 Процесс мониторинга (Monitor Process)

Соблюдение плана проверяется постоянно и через predetermined интервалы времени. Анализируются выходы (outputs) и условия завершения <задач>. Получаемые <в процессе измерений> результаты оцениваются в терминах требуемых характеристик (например, через процедуры обзора/оценки и аудита – review, audit). Затраты усилий (efforts), соблюдения расписания, стоимость к данному моменту, используемые ресурсы – все это исследуется к каждой дате оценки. При этом, оценивается и корректируется профиль рисков, а также, производится проверка удовлетворения требований качества.

Моделируются и анализируются данные измерений. Анализ расхождений (variance analysis) <плана с реальным выполнением проекта> базируется на оценке отклонений реальных данных от планируемых и ожидаемых. Такой анализ может проводиться в отношении оценки перерасхода средств (cost overrun), нарушения расписания и других важных характеристик – ограничений проекта (*прим. автора*). Часто выполняется “внешний” (например, с привлечением представителей заказчика, *прим. автора*) анализ качества и других измеряемых данных (например, анализ плотности дефектов – defect density analysis). Проводится повторное (уточняющее, *прим. автора*) выявление рисков и оценка их последствий (risk exposure and leverage), разрабатывается дерево решений, проводится моделирование (рисков и действий по их предотвращению) и другие работы – уже в контексте полученных данных. Все эти работы позволяют обнаруживать проблемы и идентифицировать исключения, основываясь на выходе за рамки приемлемых границ тех или иных параметров проекта (в частности, характеристик качества, *прим. автора*). Отчетность по результатам создается в соответствии с планом, а также, при выходе за заданные ограничения проекта или параметров отдельных его работ.

Автор хотел бы обратить внимание на то, что современная практика управления проектами, в частности, разработки и сопровождения программного обеспечения, требует обеспечения возможности доступна к актуальным данным по проекту в любой момент времени. По-сути в настоящее время возникает целый класс интегрированных инструментов и специализированных продуктов, часто называемый *project dashboard* (наиболее близкий перевод этого понятия на русский язык может звучать как “панель управления проектом”). Обычно, такие инструменты не только работают со “снимками” данных, сводя их воедино, но обращаются непосредственно к данным в системах конфигурационного управления, управления требованиями, сценариями тестирования, аудита кода, расписания проекта в соответствующих средствах управления проектами и т.п.

3.5 Процесс контроля (Control Process)

Выходы (результаты) процесса мониторинга обеспечивают базис, на основе которого принимаются те или иные решения. Изменения в проект вносятся там, где это необходимо, и где ассоциированные риски и их влияние смоделированы и могут быть управляемы (контролируемы). Эти изменения могут проводиться в форме корректирующих действий (например, повторного тестирования определенных компонент) и могут приводить к изменению плана, работ, документов и других активов проекта. При этом, важно контролировать (идентифицировать, оценивать и принимать решения) прямые и косвенные влияния любых изменений.

В некоторых случаях, это может привести к прекращению проекта. Во всех случаях необходимо проводить контроль изменений и процедур конфигурационного управления (см. предыдущую область знаний Software Configuration Management). При этом, решения должны документироваться и сообщаться (желательно, перед этим обсуждаться) всем заинтересованным сторонам, планы – корректироваться (там где это необходимо), а все необходимые данные должны заноситься в центральную базу данных проекта (см. тему 6.3 Perform the Measurement Process).

3.6 Ведение отчетности (Reporting)

Отчеты проводятся за определенный и согласованный период времени, согласуясь с планом проекта и адресуясь заинтересованным лицам (в том числе – “внешним”, со стороны заказчика). В принципе, возможны выделить две группы отчетов – по общему состоянию проекта (именно, они обычно адресованы и заказчику), а также детализированные отчеты, подготавливаемые чаще и касающиеся отдельных групп в команде проекта, отдельных работ, групп требований, функциональных модулей и т.п.

4. Обзор и оценка (Review and Evaluation)

В критических точках проекта оценивается общий (по всему проекту, *прим. автора*) прогресс в достижении установленных целей и удовлетворении требований заинтересованных лиц. Аналогично, проводится оценка (assessment) эффективности процессов, <работы> персонала, а также инструментов и методов, использованных в работах, проведенных за заданный промежуток времени.

4.1 Определение удовлетворения требованиям (Determining Satisfaction of Requirements)

Так как достижение удовлетворения пользователей является одной из наших принципиальных (и главной, *прим. автора*) целей, представляется важным периодическая и формальная оценка прогресса в данном вопросе. Такая оценка проводится при достижении определенных вех (milestones) проекта, например, при утверждении разработанной архитектуры). При этом идентифицируются отклонения от соответствующих ожиданий (планов) и проводятся необходимые действия, связанные с результатами оценки отклонений (например, по корректировке плана, *прим. автора*). Как было отмечено выше (см. тему 3.5 “Процесс контроля”), во всех случаях проводится контроль изменений и процедуры конфигурационного управления, документируются принятые решения и обеспечивается необходимая отчетность. Дополнительную информацию, также имеющую отношение к обсуждаемому вопросу, можно найти в области знаний “Тестирование программного обеспечения” (Software Testing) в темах 2.2 “Цели тестирования” (Objectives of Testing) и 2.3 “Оценка и аудит” (Reviews and Audits).

4.2 Оценка продуктивности/результативности (Reviewing and Evaluation Performance)

Периодическая оценка продуктивности специалистов, вовлеченных в проект, обеспечивает понимание того, насколько они следуют плану, и дает возможность идентифицировать вероятные проблемы (например, конфликты между членами проектной команды). Для оценки эффективности применяются различные методы, инструменты и техники. Сам процесс оценки является систематическим, а процедуры – периодическими. Все это зависит от используемых управленческих техник, применяемых методологий, организационных принципов, сложившейся культуры, то есть рассматривается, в том числе, в контексте специфики программной инженерии, дисциплины управления проектами и общего менеджмента (*прим. автора*).

5. Закрытие (Closure)

Проект закрывается/завершается (не путайте с прекращением проекта, *прим. автора*), когда все планы и процессы выполнены и завершены. На этой стадии в результатах проекта применяются критерии оценки его успешности. Когда принимается решение о закрытии проекта, выполняются действия по архивированию проектных данных, анализу результатов, полученных в процессе работы над проектом (post mortem analysis), и улучшению процессов (process improvement).

5.1 Определение <критериев> закрытия проекта (Determining Closure)

Проект закрывается, когда завершены специфицированные в плане проекта задачи и подтверждено <удовлетворительное> достижение критериев завершения (completion criteria) проекта. При этом, все запланированные результаты (продукт, его модули и т.п.) должны быть переданы заказчику и/или в эксплуатацию с приемлемыми (с точки зрения требований) и принятыми (со стороны заказчика) характеристиками. Удовлетворение требованиям – проверено и подтверждено/утверждено заказчиком, а цели проекта – достигнуты. Перечисленные процессы, в общем случае, требуют вовлечения всех заинтересованных лиц. Результаты их выполнения документируются, включая подтверждения со стороны заказчика о соответствии результатов проекта заданным требованиям

(client acceptance list, например, по результатам приемочных, или, как их еще называют, приемосдаточных тестов) и, если это необходимо, включая также отчеты об оставшихся/требующих доработки проблемах (known problems).

5.2 Работы по закрытию проекта (Closure Activities)

После того, как принято и утверждено решение о закрытии проекта (также говорят о “подтверждении закрытия/завершения проекта”) создается архив материалов в соответствии с утвержденными заинтересованными лицами методами, местоположением, формой и заданной длительностью хранения. База данных измерений <в организации> обновляется в соответствии с полученными финальными данными проекта и проводится пост-проектный анализ этих данных. Анализ по завершении проекта помогает в оптимизации процессов, практик и организационной структуры (см. область знаний Software Engineering Process).

6. Измерения в программной инженерии (Software Engineering Measurement)

Важность и роль количественных оценок - *измерений* - в управленческих практиках широко известна, растет с каждым годом и уже не раз подчеркивалась в SWEBOK. Эффективные измерения становятся одним из краеугольных камней организационной зрелости.

Ключевые термины и методы по измерениям в программной инженерии определены в стандарте ISO/IEC 15939:2002 Software Engineering - Software Measurement Process (2002 г.), основывающемся на международном словаре метрологии, выпущенном ISO в 1993 году. Несмотря на это, в различной литературе встречаются разные термины, например, часто термин “*metric*” – *метрика* (автор предпочитает использовать именно этот термин) используется вместо “*measure*” – *измерение*.

Данная тема следует указанному международному стандарту ISO/IEC 15939, который описывает процесс, определяющий действия/работы (activities) и задачи (tasks)*, необходимые для реализации процесса ведения измерений, а также включающий информационную модель измерений.

* “действия/работы” - *activities*, “задачи” – *tasks*: термин “задача” используется для более глубокого уровня детализации, чем “действия/работы”; термины “действия” и “работы”, как вы уже заметили, часто используются взаимозаменяемым образом. Так или иначе, это вопрос договоренности о терминологии при организации WBS (Work Breakdown Structure) – структуры декомпозиции работ.

6.1 Установление и поддержка процесса ведения измерений (Establish and Sustain Measurement Commitment)

- Формулируются требования в отношении измерений. Каждая попытка измерения должна руководствоваться организационными целями и следовать набору измерений, выполняемых в отношении требований, в соответствии с принятыми организационными или проектными стандартами. Например, в качестве организационной цели может выступать “выпуск на рынок новых продуктов первыми”. Это, в свою очередь, может порождать требование того, что факторы, способствующие достижению цели также должны быть оценены количественно, с тем, чтобы проект мог быть управляем в процессе достижения заданного результата. Для этого необходимо:
 - Определить содержание измерений. Необходимость принять, в каких масштабах – на уровне какой *организационной единицы** будут проводиться измерения – только в одной функциональной области, в рамках проекта, на уровне комплекса проектов или в организации, в целом. Все последующие задачи по ведению измерений, связанные с соответствующими требованиями, ведутся в рамках принятого содержания измерений. Безусловно, в дополнение к этому необходимо идентифицировать всех заинтересованных лиц, ассоциированных и вовлеченных в проведение измерений.
 - Заручиться поддержкой менеджеров и персонала в ведении измерений. Такая поддержка должна быть оформлена формально, сообщена персоналу и поддержана соответствующими ресурсами (см. ниже).

* *организационная единица* - *organizational unit*: этот термин будет использоваться достаточно часто в контексте ведения измерений для описания границ измерений. При этом подразумевается не структурная единица в организации, а единица деятельности, в отношении которой проводятся измерения – операция, задача, работа,

проект, программа, деятельность организации, в целом. *прим. автора.*

- Выделяются соответствующие ресурсы для проведения измерений. Организационная поддержка измерений является основным фактором успеха, так как назначение ресурсов просто необходимо для реализации процесса ведения измерений. Назначение ресурсов включает распределение ответственности (например, пользователь, аналитик и т.п.) в отношении различных задач процесса измерения и предоставление необходимого финансирования, обучения, инструментов и поддержки процесса сопровождения на постоянной основе.

6.2 Планирование процесса измерений (*Plan the Measurement Process*)

- Задание “организационной единицы” (в понимании, в частности “единицы деятельности”, как описывалось выше, *прим. автора*). Таким образом обеспечивается явный контекст измерений и связанные с ним ограничения. В качестве такой “единицы” (в общем случае, *прим. автора*) могут выступать организационные процессы, прикладные домены (области деятельности или отдельных работ) и т.п. См. подробное описание характеристик организационной единицы в уже упомянутом стандарте ISO 15939-02 (ISO/IEC 15939:2002 Software Engineering - Software Measurement Process, раздел 5.2.1).
- Идентификация информационных потребностей <в отношении результатов измерений>. Такие потребности, обычно, базируются на целях, ограничениях, рисках и проблемах на уровне заданной организационной единицы. В основе указанных аспектов лежат различные цели – организационные, проектные и т.п. Все эти аспекты (как и порождающие их цели) должны быть четко идентифицированы и для них должны быть определены соответствующие приоритеты. Затем, должно быть выбрано подмножество аспектов, в отношении которых будут проводиться измерения, и полученные результаты также должны быть документированы, персонал поставлен в известность о них, а заинтересованным лицам необходимо провести требуемую оценку аспектов измерений (см. стандарт ISO 15939-02, раздел 5.2.2).
- Выбор метрик (измерений). Кандидаты в метрики должны быть выбраны на основе приоритетов информационных потребностей и других критериев – таких, как стоимость сбора данных, возможность срыва процессных работ при сборе данных (например, в силу недостатка ресурсов), легкость анализа, легкость получения точных и целостных данных и т.п. (см. стандарт ISO 15939-02, раздел 5.2.3 и приложение C).
- Определение наборов <собираемых> данных, а также процедур анализа и ведения отчетности. Это включает в себя коллекцию процедур и расписаний, хранение, проверку, анализ, отчетность и конфигурационное управление собираемыми данными. (см. стандарт ISO 15939-02, раздел 5.2.4).
- Определение критериев оценки информационных продуктов (т.е. результатов измерений, *прим. автора*). На критерии оценки влияют технические и бизнес цели, сформулированные прежде для соответствующей организационной единицы. Результаты измерений* ассоциированы с создаваемым продуктом <являющемся целью проекта>, а также с процессами, обеспечивающими управление и измерения в проекте. (см. стандарт ISO 15939-02, раздел 5.2.5 и приложения D, E).

* в данной теме в отношении результатов измерений часто используется термин “информационный продукт” – information product.

- Оценка, утверждение и предоставление ресурсов для проведения измерений.

- План измерений должен быть оценен и утвержден соответствующими заинтересованными лицами. Это включает процедуры сбора данных, их хранения, анализа и отчетности; критерии оценки; расписание и распределение ответственности. Критерии обзора и оценки этих артефактов должны быть установлены на уровне организационной единицы или выше. Такие критерии должны принимать во внимание существующий опыт, доступность ресурсов и потенциальный срыв проекта когда предлагается изменение существующих (и уже

используемых, *прим. автора*) практик. Утверждение (approval) <выделения ресурсов> демонстрирует поддержку и принятие обязательств по проведению измерений. (см. стандарт ISO 15939-02, раздел 5.2.6.1 и приложение F).

- Ресурсы должны быть доступны для реализации запланированных и утвержденных задач по ведению измерений. Доступность ресурсов может быть распределена по стадиям внедрения изменений в процесс измерений, например, когда изменения производятся изначально в “пилотном” режиме, а уже затем, становятся составной частью стандартного процесса (т.е. используемого в рамках всего проекта, подразделения или организации, *прим. автора*). Также, необходимо уделять внимание ресурсам, необходимым для успешного внедрения новых процедур и измерений (метрик). (см. стандарт ISO 15939-02, раздел 5.2.6.2).

- Овладение и внедрение технологий поддержки <измерений>. Это включает оценку доступных технологий, выбор наиболее соответствующих (заданному контексту и ограничениям) технологий, их приобретение и овладение ими и, наконец, внедрение в повседневную практику. (см. стандарт ISO 15939-02, раздел 5.2.7).

6.3 Выполнение процесса измерений (*Perform the Measurement Process*)

- Интеграция процедур проведения измерений с соответствующими процессами. Процедуры измерения (например, сбор данных) должны быть интегрированы в оцениваемые процессы. Это может приводить к изменению самих процессов для адаптации действий по сбору или генерации необходимых данных. Это может подразумевать и анализ существующих процессов для минимизации дополнительных усилий, и оценку влияния на сотрудников, необходимые для реального принятия процедур проведения измерений. Важно понимать и принимать во внимание моральные и другие аспекты “человеческого фактора”, без которых проведение измерений, как дополнительная (к функциональной) деятельность будет восприниматься лишь как помеха основной работе. Более того, процедуры измерений должны обсуждаться с теми, кто непосредственно предоставляет данные; может потребоваться соответствующее обучение персонала; необходимо обеспечить и соответствующую поддержку (по аналогии с технической поддержкой программного обеспечения). Анализ данных и процедуры отчетности должны быть интегрированы в организационные и/или проектные процессы. (см. стандарт ISO 15939-02, раздел 5.3.1).
- Сбор данных. Данные должны собираться, верифицироваться и сохраняться <для дальнейшего использования>. (см. стандарт ISO 15939-02, раздел 5.3.2).
- Анализ данных и создание информационного продукта (как результата измерений, позволяющего принимать на его основе те или иные решения, *прим. автора*). Данные могут агрегироваться, трансформироваться или записываться как часть процесса анализа в соответствии с природой данных и информационными потребностями. Обычно результаты анализа представляются в форме соответствующих графиков, численных характеристик или других индикаторов, интерпретируемых и передаваемых, в конце концов, заинтересованным лицам. Результаты и сделанные на их основе заключения должны быть оценены (reviewed) в соответствии с процессом, определенным в организации (который может быть формальным или неформальным). Лица, предоставляющие данные и проводящие измерения, должны участвовать в процессе обзора и оценки (review) данных для обеспечения соответствия их содержательной стороны и точности, а также выполнения действий, обоснованных результатами последующего анализа. (см. стандарт ISO 15939-02, раздел 5.3.3 и приложение G).
- Обсуждение результатов. Полученный “информационный продукт” должен быть документирован и передан пользователям и заинтересованным лицам. (см. стандарт ISO 15939-02, раздел 5.3.4).

6.4 Оценка измерений (*Evaluate Measurement*)

- Оценка информационного продукта. Такая оценка проводится на соответствие специфицированным критериям оценки и определяет сильные и слабые стороны (strenghts and weaknesses*) полученного информационного продукта. Оценка может проводиться в рамках внутренних процессов или внешнего аудита и должна включать анализ отзывов от лиц, использующих полученные результаты. Сделанные выводы (в англоязычных источниках по оценке и совершенствованию процессов повсеместно используется термин *"lessons learned"* – *"полученные уроки"*, прим. автора) должны быть записаны в соответствующую базу данных (иногда называемую также *"базой знаний"* – *"knowledgebase"*, прим. автора). (см. стандарт ISO 15939-02, раздел 5.4.1 и приложение D).
- Оценка процесса проведения измерений. Данная оценка проводится на соответствие специфицированным критериям оценки и определяет сильные и слабые стороны самого процесса. Оценка может проводиться в рамках внутренних процессов или внешнего аудита и должна включать анализ отзывов от лиц, использующих полученные результаты. Сделанные выводы должны быть записаны в соответствующую базу данных. (см. стандарт ISO 15939-02, раздел 5.4.1 и приложение D).
- Определение потенциальных возможностей улучшения/усовершенствований (improvements) <процесса проведения измерений>. Такие рекомендации по улучшению могут заключаться в изменении формата используемых количественных индикаторов, единиц измерения или изменений в их классификации (категориях метрик). Необходимо определять стоимости и отдачу (benefits) от предлагаемых улучшений и отобрать те из них, которые соответствуют целям и критериям измерений, после чего сформулировать действия, необходимые для внедрения выбранных улучшений. Предполагаемые улучшения должны быть обсуждены и утверждены заинтересованными лицами. Отсутствие потенциальных улучшений (если они не были идентифицированы в результате проведенного анализа) также должно быть обсуждено с заинтересованными лицами.

* strenghts and weaknesses – два из четырех элементов SWOT-анализа. SWOT - Strengths, Weaknesses, Opportunities, Threats – сильные стороны, слабые стороны, возможности, угрозы. Обычно представляется как квадрант четырех указанных факторов.