

"Если даже каждый будет делать все, что в его силах, это все же не будет ответом, адекватным сложности проблемы качества".

Э.Деминг

Качество программного обеспечения

2020

Полёт на Марс 23 ноября 2016

Сбой в работе блока, измеряющего угловые скорости, произошёл при входе «Скиапарелли» в атмосферу Марса. Получив неверные данные, система управления неправильно вычислила высоту, которая оказалась отрицательной, то есть ниже уровня поверхности. Это привело к преждевременному раскрытию парашюта и срабатыванию тормозных двигателей, а затем к активации на высоте 3,7 километра системы посадки, которая должна была сработать уже после того, как модуль сел бы на Марс.

Сбой в Мегафоне 19 мая 2017

Генеральный директор компании «МегаФон» С.Солдатенков: «Технический сбой, который произошел вчера, полностью устранен».

В.Добрынин: «Мы приняли решение выплатить компенсацию до 20 млн. руб. клиентам»

Директор по связям с общественностью компании П.Лидов заявил, что причиной проблем со связью стала «ошибка софта».

Неудачный запуск с космодрома Восточный 28 ноября 2017 г.

Комиссия Роскосмоса опубликовала результаты расследования аварии.

«Поведение разгонного блока ... выявило скрытую проблему в алгоритме, которая не проявлялась десятилетиями успешных пусков».

Система управления начала выдавать воздействие на разворот орбитального блока, но «Фрегат» ушёл с курса. «Сложилось такое сочетание параметров, которое не встречалось ранее. Соответственно, не было выявлено».

Разбился Boeing 737 Max 10 марта 2019

В Эфиопии разбился Boeing 737 MAX, погибло 157 человек. Предыдущая катастрофа с таким самолётом была 29 октября 2018 г., погибло 189 человек. Согласно данным самописцев, в работе приборов лайнера было несколько сбоев.

У обоих лайнеров через несколько минут после взлета датчик угла атаки ошибочно показал, что нос судна опасно задран вверх. В результате автоматическая система предотвращения сваливания направила самолет вниз.

В корпорации Boeing ответили, что готовят обновление программного обеспечения.

Вывод денег через СБП 24 августа 2020

ЦБ выявил способ хищения средств со счетов клиентов с через Систему быстрых платежей (СБП).

Злоумышленник через уязвимость в банковской системе получил данные счетов, запустил мобильное приложение в режиме отладки, авторизовавшись как реальный клиент, отправил запрос на перевод средств в другой банк, но перед совершением перевода вместо своего счета отправителя указал счет другого клиента этого банка.

Система дистанционного банковского обслуживания (ДБО), не проверив принадлежность счета отправителю, направило в СБП команду на перевод средств, который та и осуществила.

Проблема качества программ

Возникла давно, трактовалась по-разному

- точность при ограничении на ресурсы
- надёжное хранение информации

Доступные вычислительные ресурсы – новые проблемы

- сложность взаимодействия
- слабая защищённость
- перегруженность функциями
- низкая надёжность

Что позволит улучшить качество?

Истоки инженерного подхода

Первоочередные меры по настройке системы управления качеством на предприятии (Шухарт)

- математически обоснованные требования к изготовлению каждой части изделия
- контроль – инструмент разделения системных и случайных ошибок
- оговаривается максимальное количество случайных ошибок
- совокупное количество дефектов в готовом изделии изначально учитывается в технологии производства

Управление качеством по Демингу – теория нравственности (1)

Избранные принципы Деминга (всего их 14)

- 3. Покончите с зависимостью от массового контроля
- 5. Улучшайте каждый процесс
- 6. Введите в практику подготовку и переподготовку кадров
- 7. Учредите лидерство
- 8. Изгоняйте страхи
- 12. Дайте работникам возможность гордиться своим трудом

Управление качеством по Демингу – теория нравственности (2)

Смертельные болезни и препятствия

- Никто не может преуспеть, используя только количественные критерии...
- Тот, кто управляет компанией, основываясь только лишь на точных числах, в скором времени останется и без компании, и без чисел...
- Наиболее важные для управления величины неизвестны и неопределимы количественно.

Инженерный подход (1)

Систематический, структурированный, количественный подход к разработке, функционированию и сопровождению программных средств; способ применения инженерных методов в разработке программных продуктов

Инженерный подход (2)

Сторонники подхода:

организованная группа средних специалистов со знанием правильного процесса может легко переиграть команду суперпрограммистов, ориентирующихся на мастерство

Противники подхода:

при быстрой разработке гибких, адаптируемых систем инженерные процессы непригодны, системы получаются громоздкими, неуклюжими и дорогими

Что понимается под качеством

Точки зрения

Заказчик: снята производственная проблема за невысокую цену и без дополнительных трудностей.

Пользователь: удобство работы с системой.

Конкуренция: небольшое время реализации.

Сопровождающий программист: легко адаптировать к изменяющимся условиям эксплуатации.

Качество и стандарты

- Стандарт ISO 8402:94. Качество – весь объём признаков и характеристик программ, который относится к их способности удовлетворять установленным или предполагаемым потребностям.
- ГОСТ РВ 51987–2002. Качество – совокупность свойств, обуславливающих их пригодность в соответствии с целевым назначением.
- ISO/IEC 25000:2014. Качество – способность программного продукта при заданных условиях удовлетворять установленным или предполагаемым потребностям.

Модели качества ПО

Стандарт ISO 8402:94. Используется 21 характеристика качества, которые объединены в 6 групп: функциональная пригодность, надёжность, применимость, эффективность, сопровождаемость, переносимость.

ISO/IEC 25000:2014. Модель включает 8 характеристик: функциональная пригодность, надёжность, удобство использования, производительность, сопровождаемость, переносимость, совместимость, защищённость.

Целевое назначение и пригодность

Заказчик: своевременная поставка, быстрая окупаемость, функциональная полнота, непротиворечивость информации, приемлемая производительность, надёжность, защищённость.

Пользователь: наличие эксплуатационной документации, удобство взаимодействия, снижение сложности выполняемых операций, уменьшение утомляемости.

Сопровождающий программист: единый стиль программного кода, нормализованная база данных, тестируемость, модифицируемость.

Специалисты о качестве (1)

Р. Гласс

- переносимость
- надёжность
- эффективность
- удобство работы
- тестируемость
- понятность
- модифицируемость

А. Купер

качество – простота
взаимодействия пользователя
с системой

Дж. Рейнвотер

программное изделие
качественное, если
пользователь им удовлетворён

Специалисты о качестве (2)

А.Н.Терехов

изделие некачественное, если

- неадекватно функционирует,
- есть отказы при применении по назначению,
- нет эффективности по времени отклика,
- нарушается конфиденциальность информации,
- есть несоответствие хранимых данных и вводимой информации.

Измерение качества (1)

Для оценки и сравнения характеристик качества требуются измерительные алгоритмы на основе объективных критериев.

Стандарты рекомендуют предусмотреть возможность объективного и воспроизводимого измерения каждой из характеристик качества для сопоставления с техническим заданием с учётом нормы допустимых ошибок измерения.

В этом случае необходима процедура определения числовых значений параметров проекта.

Измерение качества (2)

Характеристики качества делятся на три уровня детализации показателей:

- количественные, которые можно измерить и численно сопоставить с требованиями;
- категорийные, которые представляются шкалой свойств;
- качественные, которые характеризуются порядковой или точечной шкалой (есть — нет, хорошо — плохо) и определяются экспертным методом.

Количественные характеристики (1)

Определяются непосредственно, поэтому их допустимые значения следует задать ещё при проектировании системы, а лучше — на стадии анализа требований.

Они могут быть представлены численными значениями, согласованными с заказчиком и пользователем.

Таких характеристик немного: эффективность, которая может быть зафиксирована в ТЗ, и, с некоторыми оговорками, надёжность.

Количественные характеристики (2)

К измеряемым характеристикам относят и такие метрики, как объём текста программы, его сложность, сложность по управлению, объектно-ориентированные и интегральные метрики.

Их вычисление формально определено и может быть выполнено автоматически специальными средствами.

Однако они составляют незначительную долю исследований на качество, хотя время на них возлагали большие надежды.

Польза от них сомнительная: удовлетворительный код может содержать некорректные функции, не соответствовать требованиям, неудовлетворительно обрабатывать исключения.

Количественные характеристики (3)

Надёжность функционирования системы определяется, в основном, её безошибочностью: программные элементы не подвержены физическому износу, и, будучи изначально правильными, не портятся от времени.

Почти всегда некорректность функционирования — следствие ошибок, не выявленных во время тестирования.

Но может быть ситуация, которая идентифицируется пользователем как ошибка, но на момент разработки она ошибкой не была: после сдачи системы в эксплуатацию требования изменились.

Ошибки

Ошибки условно можно разделить на две категории: ошибки проектирования и ошибки реализации.

Первый тип ошибок выявляется тестированием системы на соответствие ТЗ. Их наличие говорит о нарушении функциональной пригодности.

Ошибки *второго* типа определяют надёжность, которая измеряется степенью покрытия тестами программных компонентов и системы в целом.

Мерой может служить относительное количество протестированных функций и маршрутов, но при этом следует учитывать уровень потенциальной опасности отдельных элементов, оценка которого часто сложна и не однозначна.

Удобство взаимодействия

Трудно оценить такую характеристику, как удобство взаимодействия.

В различных работах и стандартах приводятся примеры «хороших» и «плохих» пользовательских интерфейсов, делаются попытки определить правила их построения, но корректных критериев количественной оценки их качества нет.

Иногда требования к пользовательскому интерфейсу с той или иной степенью детализации приводят в техническом задании. В этом случае оно и служит критерием качества, но количественная мера соответствия всё равно не определяется.

Защищённость (1)

Защищённость системы — одна из наиболее трудно формализуемых характеристик качества, хотя это исключительно важная характеристика. То, что недавно казалось надёжно защищённым, вдруг взламывается.

Существуют стандарты по защите информационных систем, но методы взлома развиваются быстро.

Злоумышленник нередко использует такую уязвимость, как программные ошибки. Риск нарушения целостности информации уменьшается за счёт тщательного тестирования.

Полезно проводить тестирование безопасности, при котором тестировщик играет роль злоумышленника.

Защищённость (2)

Оценка безопасности определяется экспертным, регистрационным или расчетным путем.

В стандартах приводится метод оценки защищенности от опасных воздействий, конфиденциальности, защищенности от несанкционированного доступа (НСД).

В первом случае оценка — это вероятность сохранения конфиденциальности информации в течение заданного периода, во втором — вероятность отсутствия опасного воздействия в течение заданного периода, в третьем — вероятность сохранения защищенности ресурсов.

Комплексная оценка формируются на основе частных показателей. Это может быть полиномиальная или линейная зависимость с весовыми коэффициентами.

Другие характеристики

Такие характеристики, как сопровождаемость и переносимость количественно практически не измеримы, они оцениваются качественным способом.

Можно говорить, например, о трудоёмкости внесения изменений при сопровождении, но она зависит не только от качества программного кода, но и от сложности изменения, и от степени его отклонения от базового проекта, и от уровня квалификации сопровождающего программиста.

Поэтому используются экспертные оценки.

Шкалы качественных оценок

<i>Характеристики качества</i>	<i>Мера</i>	<i>Шкала</i>
<i>Изучаемость</i> трудоемкость изучения применения; продолжительность изучения; объем эксплуатационной документации; объем электронных учебников.	Чел.—часы Часы Страницы Кбайты	1—100 1—1000 10—1000 100—10000
<i>Удобство для анализа</i> стройность архитектуры программ; унифицированность интерфейсов; полнота и корректность документации.	Порядковая	
<i>Модифицируемость</i> трудоемкость подготовки изменений; длительность подготовки изменений.	Чел.-часы Часы	1—1000 1—1000

Управление качеством (1)

Управление качеством – действия, направленные на улучшение качества продукции.

Понятие качества нечёткое, его обеспечение тоже.

В стандарте ИСО 8402:1994 под управлением качеством понимаются «методы и виды деятельности оперативного характера, используемые для выполнения требований к качеству».

В ГОСТ 15467–69 управление качеством продукции определяется как «действия, осуществляемые при создании и эксплуатации или потреблении продукции в целях установления, обеспечения и поддержания необходимого уровня её качества».

Управление качеством (2)

В 1987 году на базе национальных стандартов качества были разработаны и опубликованы международные стандарты ИСО серии 9000 с рекомендациями по созданию систем качества.

Основная цель заключалась в том, чтобы заказчик, зная, что поставщик использует эту систему, был уверен в качестве работы программного обеспечения.

В результате накопленного опыта применения этих стандартов последовательно, в 1994, 2000, 2008 и 2015 годах, выходили новые их редакции.

Управление качеством (3)

Стандарт ИСО 9000:2005 определяет управление качеством как «часть менеджмента качества, направленная на выполнение требований к качеству».

Под менеджментом качества понимаются аспекты управления, которые определяют политику в области качества, цели и ответственность, планирование качества, его обеспечение и улучшение в рамках системы качества.

Объект управления качеством — процесс создания продукции. Субъектами управления качеством выступают руководители всех уровней, которые воздействуют на процесс создания продукции. Вместе они образуют систему управления качеством.

Управление качеством (4)

Для достижения качества в компании необходимо организовать процедуру обеспечения качества и создать соответствующую структуру: группу оценки качества.

Основные функции подобных структур — комплексная проверка всех параметров качества в процессе выполнения проекта и особенно перед выпуском продукта.

Обнаруженные дефекты имеют различную степень тяжести, в зависимости от этого либо продукт возвращается на доработку, либо блокируется ошибочная функция, либо вносятся косметические правки, либо ничего не делается в надежде, что пользователь его примет.

Управление качеством (5)

Проверка должна состоять из двух частей: контроль параметров на соответствие формальным требованиям и контроль содержания.

В первой части проверяется, все ли артефакты, подлежащие проверке, в наличии и правильно ли они оформлены.

Вторая подразумевает проверку собственно качества, включая неформальные параметры.

Однако, если качество разработчику не нужно, проверка становится пустой формальностью.

Риски качества (1)

Разделим риски качества условно на четыре группы:

- 1) внешние, объективные;
- 2) проектные, но не связанные с разработчиками;
- 3) связанные с разработчиками, но не с программистами;
- 4) связанные с программистами.

Объективные риски

Связаны с событиями, внешними по отношению к проекту:

- появление на рынке конкурирующих продуктов,
- появление новой технология,
- неожиданные правительственные решения, например, введение или изменение законов, вынуждающих срочно изменять техническое задание.

Проектные риски

В основном, ошибки в управлении проектом.

Например, известно, когда будет противостояние Земли и Марса, легко вычисляется время старта корабля на Марс, следовательно, и срок, к которому должно быть готово программное обеспечение.

Примерно можно определить период разработки качественного программного обеспечения.

Но если сроки назначаются нереальные, корабль, немного полетав вокруг Земли, падает в океан.

Риски, связанные с разработчиками

Возникают на этапах, предшествующих программированию.

Особо тяжёлый случай – некачественно проведённый анализ требований. Причины могут быть разными, результат печальный: качеству будет нанесён непоправимый урон.

К значительному снижению качества приводит неудачное проектирование данных.

Критичная проблема – взаимодействие процессов, особенно в реальном времени.

Риски программирования

Области риска:

1. Квалификация программиста
2. Сложность программы (объективная сложность, беспорядочный код, плохой стиль, неверное использование прототипа)
3. Некачественная отладка.
4. Переход на новые методы разработки.
5. Некорректная обработка нештатных ситуаций.
6. Неверная оценка компонентов при компонентном подходе.

Системы качества (история)

Комплексные системы управления качеством продукции формировались в США, Европе, Японии, Советском Союзе в 1970-е годы.

Начало их внедрения – Саратовская система бездефектного изготовления продукции (середина 1950-х годов).

В 1970-х годах во Львове была разработана комплексная система управления качеством продукции (КС УКП). Она позволила объединить элементы управления качеством в единую систему.

Модель технологической зрелости СММИ (Capability Maturity Model Integration)

Версия 2.0 в 2018 годурия.

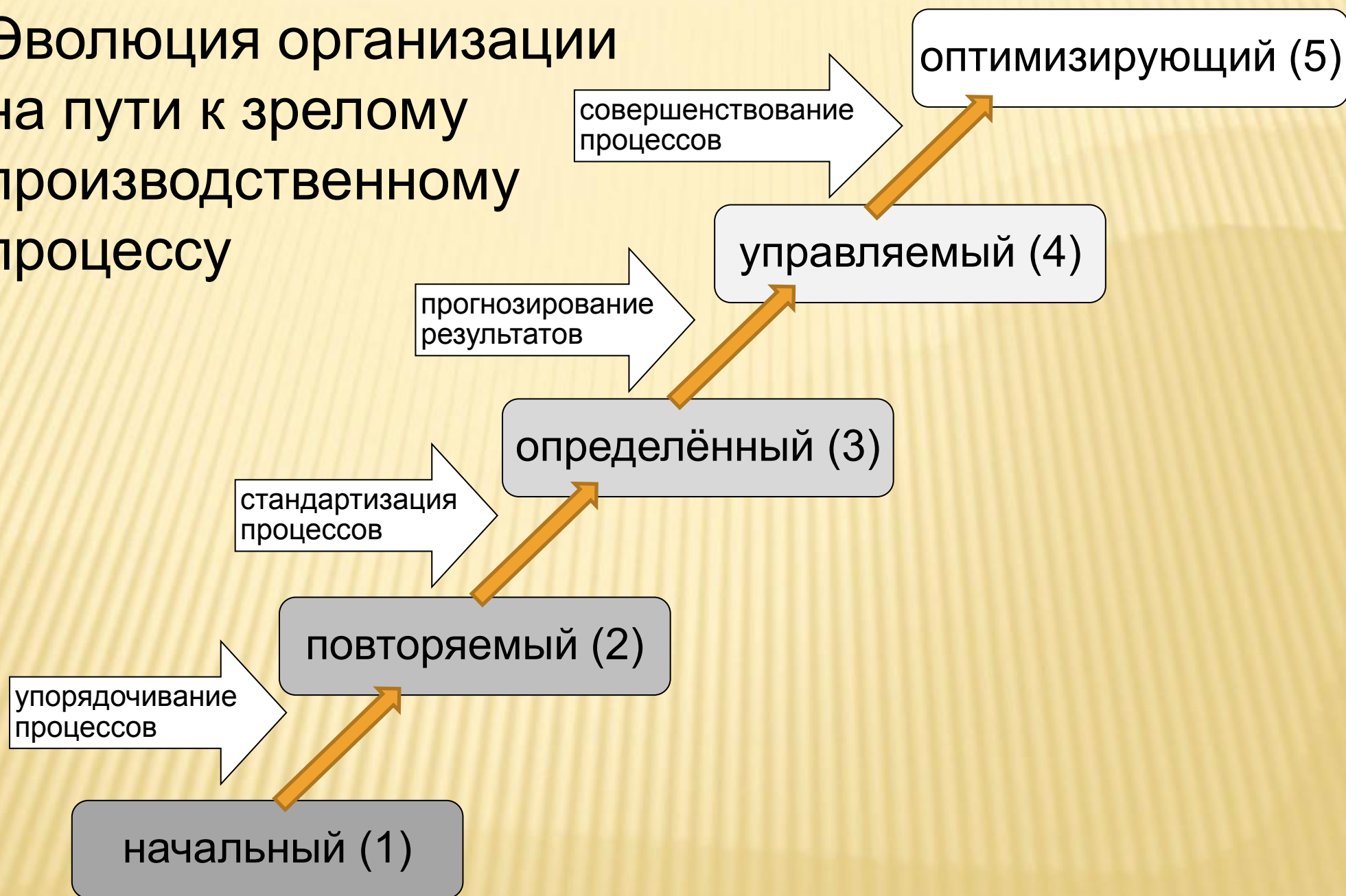
Цель –

определить надёжных разработчиков
программного обеспечения

Профессиональная зрелость коллектива

характеризуется прочными связями между
участниками, которые возникают на основе общих
ценностных ориентаций, позитивных
неформальных отношений

Эволюция организации на пути к зрелому производственному процессу



Уровень 1 – начальный

Процессы носят случайный характер. Успех проекта зависит только от руководителя и исполнителя. Работа представляется как черный ящик.

Ключевых процессов нет

Уровень 2 – повторяемый

Есть контрольные точки внутри процесса. Процессами можно управлять по аналогии. Вводится понятие эффективности. Работа представляется как цепочка черных ящиков.

Группы ключевых процессов

- 1) управление требованиями
- 2) планирование проекта
- 3) мониторинг и контроль проекта
- 4) управление контрактами
- 5) обеспечение качества процесса и продукта
- 6) управление конфигурацией

Уровень 3 – определённый

Процессы документируемые, представляют собой единую технологическую среду. В проектах используется адаптированная версия стандартного производственного процесса.

Группы ключевых процессов

- 7) определение производственного процесса
- 8) координация производственного процесса
- 9) обучение
- 10) интегрированное управление разработкой
- 11) инженерия разработки
- 12) межгрупповая координация
- 13) экспертные оценки

Уровень 4 – управляемый

Собираются количественные показатели производственного процесса и качества продукта. Они оцениваются и контролируются с количественной точки зрения. Все процессы проходят стабильно.

Группы ключевых процессов

14) количественное управление процессом

15) управление качеством программного обеспечения

Уровень 5 – оптимизирующий

Количественная обратная связь с процессом.
Реализация передовых технологий, проверка их на пилотных проектах и применение в остальных проектах. Выявляются дефекты, оцениваются их причины и предупреждается их появление.

Группы ключевых процессов

- 16) предотвращение дефектов
- 17) управление технологическими изменениями
- 18) управление изменениями процесса

Недостатки СММІ

- Модель управляет проектом, а не процессом создания программного продукта;
- нет анализа рисков, невозможно своевременно обнаружить проблемы;
- для небольших компаний модель слишком бюрократична.

Методика критически важных практических навыков СВР

Лучший практический навык тот, который решит наибольшее число проблем

Цели методики

- Внедрить лучшие навыки разработки ПО;
- тратить ресурсы на разработку, а не на формальное следование инструкциям;
- использовать лучшие практические навыки с учётом корпоративной культуры;
- дать возможность эффективно учиться методикам.

Методика СВР

Общие рекомендации

- Делать упор на коллективный характер завершения разработки;
- придумать быструю стратегию реализации;
- измерять продвижение к цели;
- измерять активность разработки.

Методика СВР

Принципы управления проектом

- Ошибки выявляются и исправляются сразу;
- планирование проводится на основе правильных показателей;
- минимизируется количество неконтролируемых изменений;
- эффективно используются сотрудники.

Методика СВР

Лучшие навыки

1. Формальное управление рисками.
2. Соглашение об интерфейсе.
3. Формальная проверка проектов.
4. Управление проектами на основе метрик.
5. Контроль качества продукта на глубоком уровне.
6. Общедоступная информация о ходе проекта.
7. Исправление причин ошибок для достижения качества.
8. Управление конфигурацией.
9. Управление персоналом.

Экстремальное программирование (XP)

Базовые принципы

- быстрая обратная связь;
- приемлемая простота;
- постепенное изменение;
- приемлемое изменение;
- качественная работа.

Виды деятельности

- кодирование;
- тестирование;
- слушание (внимание);
- проектирование.

Экстремальное программирование (XP)

Двенадцать правил

1. Краткосрочное планирование.
2. Небольшие версии.
3. Метафора.
4. Простой дизайн.
5. Тестирование.
6. Переработка кода.
7. Программирование парами.
8. Коллективное владение.
9. Постоянная интеграция.
10. 40-часовая рабочая неделя.
11. Заказчик на месте разработки.
12. Стандарты кодирования.

Экстремальное программирование (XP)

Механизмы обеспечения качества

- Уделять внимание только тем вопросам, которые в данный момент действительно важны;
- добиваться получения заказчиками и руководителями оперативной информации о результатах работы и о процессах.

Методология Scrum

Цель —
максимально быстрая поставка
продукта на рынок.

Для этого используется понятие
«минимально ценный продукт»
(*minimum viable product*, MVP)

Методология Scrum

Аргументы

- рынок должен получить данный продукт до того, как тот успеет устареть;
- до выпуска продукта на рынок есть лишь смутные предположения о потребностях пользователя, и только выпуск продукта на рынок даст возможность их уточнить.

Проблемы решаются качественным анализом, но в Scrum его нет

Методология Scrum

Жизненный цикл разработки разбивается на спринты, каждый из которых длится от одной недели до месяца.

Цель спринта —
выпуск работающей версии продукта, реализующей какую-то функциональность будущей системы.

Методология Scrum

Качество

- Время спринта оценивается произвольно: нет стабильно качественной отладки;
- нет полноценного анализа: ошибки в функциональности;
- накапливается технический долг.

Вывод: для производства качественных продуктов методология не пригодна.

Литература

- Гласс Р. Программирование и конфликты 2.0
- Демарко Т., Листер Т. и др. Балдеющие от адреналина и зомбированные шаблонами. Паттерны поведения проектных команд.
- Стандарты: ГОСТ Р ИСО/МЭК 25010–2015 Информационные технологии. Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Модели качества систем и программных продуктов.
- Стандарты: ГОСТ Р ИСО/МЭК 9126–93. Оценка программной продукции. Характеристики качества и руководства по их применению.
- Стандарты: <http://www.gost.ru/>, обновления и новые стандарты: <http://protect.gost.ru/>.
- Стандарты: ISO/IEC 25000:2014(en) Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Guide to SQuaRE