

Методология функционального моделирования. – М.: ИПК Издательство стандартов, 2001. – 50 с.

3. Кане М.М. Системы, методы и инструменты менеджмента качества: учебное пособие / М.М. Кане, Б.В. Иванов. – СПб.: Питер, 2008. – 560 с.

4. Технология вакуумной ионно-плазменной обработки: учебное пособие / В.В. Будилов, Р.М. Киреев, С.Р. Шехтман. – М.: Изд-во МАИ, 2007. – 155 с.

5. Черемных С.В. Моделирование и анализ систем. IDEF-технологии: практикум / С.В. Черемных, И.О. Семенов, В.С. Ручкин / М.: Финансы и статистика. – 2006. – 192 с.

Structurally Functional Model of the Technological Process of Coating Deposition Obtained by Vacuum-arc Discharge

Yansaitova M.I., graduate student of the Ufa aviation technical university; Republic of Bashkortostan, Ufa

e-mail: denik545@ya.ru

Summary. In this work technological process of drawing the coverings received by sedimentation from the vacuum and arc category on the basis of creation of structurally functional model is considered. Operations of technological process of drawing the coverings received

by sedimentation from the vacuum and arc category are united in 3 stages and each stage is in detail described, using methodology of modeling of IDEF0.

Keywords: technological process, coatings obtained by vacuum-arc discharge, controlled parameters, IDEF0 modeling methodology.

References:

1. Budilov V.V., Yagafarov I.I., Yansaitova M.I. Reserch of dependence of microhardness and phase structure of a covering of TiN on an arrangement of details in the vacuum chamber at sedimentation from plasma of the vacuum and arc category. *The strengthening technologies and coverings*. 2017, Volume 13. No. 1. pp. 20–23.

2. State Standard 50.1.028-2001 Information technologies of support of life cycle of production. Methodology of functional modeling. *Publishing and printing complex «Standards Publishing House»*. Moscow, 2001. 50 p.

3. Kane M.M., Ivanov B.V. Systems, methods and instruments of quality management: manual. *Piter*. St-Petersburg, 2008. 560 p.

4. Budilov V.V., Kireev R.M., Shekhtman S.R. Technology of vacuum ion-plasma processing: manual. *Publishing house of the Moscow aviation institute*. Moscow, 2007. 155 p.

5. Cheremnykh S.V., Semyonov I. O., Ruchkin B.C. Modeling and analysis of systems. IDEF technologies: practical work. *Finance and statistics*. Moscow, 2006. 192 p.

Реализация процесса тестирования в Agile-методологиях

О.В. Ерина

исполнительный директор
ООО «Спейс-О Технологии»; г. Томск

e-mail: olga@quickquality.ru

Аннотация. В статье приведен анализ реализации процессов тестирования в Agile-методологиях: *Agile Modeling, Agile Unified Process, Agile Data Method, Essential Unified Process, Extreme Programming, Feature Driven Development, Getting Real, Open UP, Scrum, Kanban*. Проанализирован мировой опыт в области организации процесса управления качеством программного обеспечения в рамках Agile. На основе общих для всех Agile-подобных характеристик был составлен и аргументирован перечень техник тестирования, подходящих всем гибким методологиям разработки программного обеспечения.

Ключевые слова: гибкие методологии разработки, тестирование программного обеспечения, качество программного обеспечения.

Введение

Процессы разработки программного обеспечения (ПО), основанные на гибкой методологии разработки ПО *Agile*, становятся очень популярными среди компаний-разработчиков ПО. Основные принципы гибкой методологии изложены в Манифесте *Agile* [1]. Но, как будет рассмотрено ниже, Манифест *Agile* не дает явных указаний по организации процесса тестирования в проекте, применяющем гибкую методологию разработки. Разработка требуемых указаний предполагает изучение имеющихся наработок по теме и проведение собственного анализа процесса управления качеством в рамках гибких методологий разработки.

В данной статье рассмотрим следующий научный вопрос: «Как организовать процесс тестирования в рамках гибкого процесса разработки с учетом имеющихся мировых исследований по данной теме?»



Методология *Agile* с точки зрения управления качеством ПО

Рассмотрим принципы *Agile* с точки зрения процесса тестирования.

1. Наивысшим приоритетом является удовлетворение потребностей заказчика, благодаря регулярной и ранней поставке ценного программного обеспечения.

Очевидно, что самым главным условием удовлетворения потребностей заказчика является производство программного обеспечения с наивысшим уровнем качества в заданных рамках сроков и бюджета, что является основной задачей тестирования.

2. Изменение требований приветствуется даже на поздних стадиях разработки.

Данный принцип означает, что в методологии *Agile* тестирование должно гибко реагировать на любые изменения в требованиях и, как следствие, стоит избегать методик тестирования, требующих фиксации требований, например, таких как рецензирование.

3. Работающий продукт следует выпускать как можно чаще, с периодичностью от пары недель до пары месяцев.

Данный принцип подтверждает мысль предыдущего пункта: если работающий продукт должен выпускаться часто, соответственно, команде, следующей принципам гибкой методологии, следует применять «легковесные» методики тестирования или методики предотвращения дефектов.

4. На протяжении всего проекта разработчики и представители бизнеса должны ежедневно работать вместе.

Для успешной реализации проекта необходимо, чтобы в переговорах участвовали три стороны: представитель заказчика, представитель программистов и представитель команды тестирования [2].

5. Над проектом должны работать мотивированные профессионалы. Чтобы работа была сделана, создайте условия, обеспечьте поддержку и полностью доверьтесь им.

Мотивации персонала (в том числе и сотрудников отдела тестирования) посвящены многие труды [3], т.к. только заинтересованный в результате специалист сможет обеспечить наивысший уровень качества результата своей работы.

6. Непосредственное общение является наиболее практичным и эффективным способом

обмена информацией как с самой командой, так и внутри команды.

Данный пункт сильно перекликается с п. 4 и означает, что специалисты из команды тестирования (или их представитель) обязательно должны присутствовать на проектных совещаниях и участвовать в принятии проектных решений.

7. Работающий продукт – основной показатель прогресса.

Как было сказано в п. 1, работающий продукт – определяющий фактор удовлетворенности заказчика работой команды-исполнителя. Задача тестирования состоит в том, чтобы выявить, что наиболее важно для заказчика, и подготовить тесты, доказывающие, что именно это он и получает [2].

8. Инвесторы, разработчики и пользователи должны иметь возможность поддерживать постоянный ритм бесконечно.

Поддержка заданного ритма на протяжении всего проекта возможна только при грамотном планировании работ, а также при применении методик предотвращения ошибок, которые будут рассматриваться далее.

9. Постоянное внимание к техническому совершенству и качеству проектирования повышает гибкость проекта.

Данный принцип говорит о том, что качество производимого ПО является одной из приоритетных областей приложения усилий со стороны как руководства проектом, так и всех членов команды.

10. Простота – искусство минимизации лишней работы – крайне необходима.

Данный принцип подтверждает, что в *Agile* предпочтительно использование «легковесных» методик тестирования.

11. Самые лучшие требования, архитектурные и технические решения рождаются у самоорганизующихся команд.

Самоорганизующиеся группы – это сотрудники, которые не нуждаются в контроле и руководстве со стороны вышестоящего менеджера. Данный принцип подразумевает, что распределение задач и контроль результата производится самостоятельно участниками процесса, а именно – разработчиками, тестировщиками, аналитиками и т.д. Такой подход возможен при условии качественного превентивного планирования, а также при высоком уровне мотивированности участников процесса, поэтому данный принцип перекликается с п. 5, гласящим, что положительная мотивация

сотрудников – мощный инструмент повышения качества работы всей команды.

12. Команда должна систематически анализировать возможные способы улучшения эффективности и соответственно корректировать стиль своей работы.

Команда тестирования, как часть команды разработки должна систематически искать пути совершенствования процесса тестирования, тесно взаимодействовать с программистами и заказчиком, помогать заказчику формулировать свои требования и искать лучшие пути обеспечения качества, созданного по требованиям, функционала системы и нефункциональных характеристик.

Отметим, что в данном манифесте ничего не сказано в явном виде об обеспечении качества производимого ПО. Если рассмотреть наиболее популярные процессы разработки, основанные на принципах *Agile*, то можно убедиться, что в них процесс тестирования также описан либо весьма поверхностно, либо не описан вообще (табл. 1).

Таблица 1.

Наличие описания процесса тестирования в *Agile*-методологиях

Методология	Описан ли процесс тестирования
<i>Agile Modeling</i>	–
<i>Agile Unified Process</i>	–
<i>Agile Data Method</i>	–
<i>Essential Unified Process</i>	–
<i>Extreme Programming</i>	+
<i>Feature Driven Development</i>	+
<i>Getting Real</i>	–
<i>Open UP</i>	+
<i>Scrum</i>	–
<i>Kanban</i>	–

Тем не менее, многие пункты манифеста предполагают, что процесс тестирования на проекте, в котором применяется методология *Agile*, должен быть продуманным и понятным всем членам команды.

Организация процесса тестирования в *Agile*-процессе

Вопрос тестирования ПО в рамках гибких методологий слабо освещен в мировой литературе, но, тем не менее, обратимся к существующему мировому опыту.

Э. Баах в своей книге «*Agile Quality Assurance*» [4] подробно рассматривает особенности процесса контроля качества ПО в рамках методологии

Agile. А именно, по мнению Э. Бааха, QA-специалисты должны:

1) мыслить масштабно и понимать QA не только как «*testing*», но и как верификацию всех процессов проекта, в частности, на соответствие принципам *Agile* (либо другой методологии, используемой командой-разработчиком), выступая в качестве *agile*-инспектора;

2) оценивать требования и документацию, чтобы убедиться, что в документах отсутствует двойственность понимания и все требования ясны;

3) оценивать пользовательские истории (*user stories*) и убедиться, что они совпадают с бизнес-требованиями и имеют достижимые критерии приемки;

4) участвовать в написании автотестов;

5) работать с рисками, которые могут возникнуть со стороны QA;

6) стремиться к предупреждению дефектов, используя современные техники/методологии (например, *TDD*).

Ф. Хэйер в своей книге «*Agile testing*» [5] акцентирует внимание читателя на таких уже известных техниках и методологиях как квадранты тестирования и *TDD*.

Л. Криспин, Дж. Грегори в книге «Гибкое тестирование. Практическое руководство для тестировщиков ПО и гибких команд» также задались вопросом оптимального управления качеством с помощью участия QA-специалистов в максимально возможном количестве процессов проекта и с применением техники квадрантов тестирования [2].

После изучения имеющегося мирового опыта рассмотрим вопрос самостоятельно и выделим основные характеристики *Agile*-процессов.

1. Итеративность процесса. При этом обязательным условием является создание работающего прототипа системы на каждой итерации.

2. Переменчивость требований.

3. Фокус на техническом совершенствовании системы.

4. Тесное взаимодействие всех членов команды.

Теперь рассмотрим, какие из уже существующих решений управления качеством ПО могут быть успешно применены в рамках процессов *Agile*. В табл. 2 показано, какие из методик тестирования обеспечат выполнение основных характеристик *Agile*:

Следует отметить, что процесс тестирования должен быть четко спланирован, т.к. наличие плана особенно необходимо в условиях неопределенности и меняющихся требований. На этапе планирования (который должен быть реализован в первой итерации) осуществляются действия, направленные на определение целей тестирова-

Методики тестирования

	Интерактивность	Переменчивость требований	Техническое совершенствование системы	Тесное взаимодействие команды
<i>TDD</i>	+			+
<i>FMEA</i>	+		+	
<i>PRAM, SST, PRisMa</i>		+	+	+
Квадранты тестирования	+	+	+	+

ния, и описание задач тестирования для достижения данных целей [3].

Во-вторых, на каждом этапе/уровне/итерации разработки ПО должен осуществляться мониторинг прогресса по процессу тестирования (например, посредством метрик процесса тестирования и метрик продукта).

В-третьих, необходимо применять «легковесные» методики тестирования, основанные на рисках. Например, такие как:

- *Pragmatic Risk Analysis and Management (PRAM)*;
- *Systematic Software Testing (SST)*;
- *Product Risk Management (PRisMa)* [3].

Применение данных техник позволяет предотвратить появление дефектов в создаваемом продукте и, как следствие, сократить расходы производства без ущерба качеству создаваемого ПО.

В-четвертых, следует использовать статические и динамические методы предотвращения дефектов, например следующие:

1. *TDD* (описанное в экстремальном программировании) – *Test Driven Development* (разработка через тестирование) – это техника разработки программного обеспечения, которая основывается на повторении очень коротких циклов разработки: сначала пишется тест, покрывающий желаемое изменение, затем пишется код, который позволит пройти тест, и наконец проводится рефакторинг нового кода к соответствующим стандартам [6]. Т.е. процесс *TDD* выглядит следующим образом:

- как только сформулированы требования к компоненту/функции системы, программист создает юнит-тест, проверяющий работу данного компонента/функции, в этот момент тесты не будут проходить успешно, т.к. код компонента еще не написан;
- далее непосредственно реализуется код компонента/функции и запускается тест, на теперь уже в написанном и соответственно существующем коде компонента/системы;
- исправление ошибок осуществляется до тех пор, пока юнит-тест не завершается успешным результатом.

TDD можно отнести к динамическим видам методов предупреждения дефектов.

2. *FMEA* – это методология проведения анализа и выявления наиболее критических шагов производственных процессов с целью управления качеством продукции [7].

Суть процесса *FMEA* заключается в сборе и анализе потенциальных дефектов и принятии мер по предупреждению данных дефектов. Сбор информации о потенциальных угрозах может быть осуществлен посредством следующих инструментов:

- привлечение экспертов в области риск-менеджмента;
- использование стандартных шаблонов из базы рисков;
- анализ ретроспективы проектов;
- «мозговой штурм».

После выявления потенциальных угроз менеджменту проекта следует сконцентрировать усилия в местах скопления данных угроз. Например, выделить больше времени на тестирование данных областей потенциального скопления дефектов либо обратиться к автоматизации для увеличения объема покрытия кода в потенциально «проблемных» областях.

FMEA можно отнести к статическим видам методов предупреждения дефектов.

Любой проект следует завершать регламентированными действиями:

- проверка, что запланированные результаты достигнуты;
- закрытие отчетов о дефектах (инцидентах) или внесение изменений в записи по каждому из «открытых» дефектов (инцидентах);
- завершение и архивирование тестового обеспечения, тестового окружения и инфраструктуры тестирования для последующего использования;
- анализ полученных уроков для определения изменений, необходимых для будущих проектов;
- использование собранной информации для повышения зрелости процесса тестирования.

Отметим, что для показательности эффективно-го управления качеством ПО необходимо использо-

вание метрик процесса (метрики риска и метрики прогресса) и метрик продукта (метрики дефектов).

Заключение

Несмотря на ограничения в применении тех или иных методов тестирования в рамках *Agile* процесса, описанных выше, команда вправе применять любые из существующих методов тестирования, как статических, так и динамических, если сможет адаптировать их в рамках гибкой методологии без потери качества конечной продукции.

Как было сказано выше, методология *Agile* набирает обороты популярности, но компаниям-разработчикам ПО следует тщательнее продумывать процесс тестирования, т.к. от него напрямую зависит качество производимого ПО. Несмотря на слабую освещенность процесса тестирования ПО в методологии *Agile*, современные методы тестирования позволяют «скомпоновать» успешную методологию управления качеством ПО.

Литература

1. URL: <http://agilemanifesto.org/iso/ru/manifesto.html>.
2. Криспин Лайза. Гибкое тестирование. Практическое руководство для тестировщиков ПО и гибких команд / Криспин Лайза, Грегори Джанет. – М: Изд-во Вильямс, 2016. – 464 с.
3. ISTQB Syllabus Advanced Level, 2012.
4. «Agile Quality Assurance. Deliver Quality Software – Providing Great Business Value» by Anthony Baah, 2016.
5. «Agile testing. An Overview» by Florian Heuer, 2014.

6. URL: <http://agiledata.org/essays/tdd.html>.
7. URL: http://www.kpms.ru/Implement/Qms_FMEA.htm.

Realization of the Testing Process in Agile-methodology

O.V. Erina, Executive director of LLC «Speys-O Tekhnologii»; Tomsk

e-mail olga@quickquality.ru

Summary. In article the analysis of implementation of processes of testing is provided in Agile-methodologies: Agile Modeling, Agile Unified Process, Agile Data Method, Essential Unified Process, Extreme Programming, Feature Driven Development, Getting Real, Open UP, Scrum, Kanban. International experience in the field of the organization of process of quality management of the software within Agile is analyzed. On the basis of the general for all Agile-like characteristics the list of the techniques of testing suitable for all flexible methodologies of software development was made and reasoned.

Keywords: flexible methodologies of development, testing of the software, quality of the software.

References:

1. <http://agilemanifesto.org/iso/ru/manifesto.html>.
2. Crispin Laisa, Gregory Janet Flexible testing. Practical guidance for software testers and flexible commands. *Publishing house Williams*. Moscow, 2016. 464 p.
3. ISTQB Syllabus. *Advanced Level*. 2012.
4. Baah A. Agile Quality Assurance. Deliver Quality Software. Providing Great Business Value. 2016.
5. Heuer F. Agile testing. An Overview. 2014.
6. <http://agiledata.org/essays/tdd.html>.
7. http://www.kpms.ru/Implement/Qms_FMEA.htm.