

Automated Software Development Process Audit

Vadim Savkin
CQG
email: vadim@cqg.com

Abstract

Internal software development process audit is a necessary instrument for ensuring high quality and stability of the process in software development companies. Goals of such audit are: a) ensure process compliance; b) assess software development process efficiency. Such audit is a rather expensive procedure. When it's automated it becomes much cheaper and gives possibility to run the procedure more frequently that leads to higher process quality and as a result higher software product quality.

Automated process audit is possible only if all valuable aspects of the process that need to be analyzed and controlled are collected in an integrated database with consistent way. Collected data must reflect real software development process with needed accuracy and details. Data collection procedures must guarantee correctness and consistency of the collected data. Experience shows that about 90% of the audit checks and more can be automated if data coverage for the process is good enough.

Keywords: internal process audit; metrics; dashboard; process data collection.

Автоматизированный контроль процесса разработки ПО

Вадим Савкин
CQG
email: vadim@cqg.com

Abstract (Russian)

Контроль процесса разработки ПО является необходимым инструментом поддержания высокого качества и стабильности процесса разработки ПО в компании. Целями такого контроля являются: а) обеспечение соблюдения заданного процесса разработки; б) оценка эффективности процесса разработки. Такой контроль является довольно дорогостоящей процедурой. Её автоматизация позволяет значительно снизить стоимость таких процедур, а значит, даёт возможность проводить их чаще, что способствует более высокому качеству процесса разработки ПО и, как результат, более высокому качеству разрабатываемых продуктов.

Автоматизация контроля процесса возможна при условии, что данные обо всех значимых аспектах процесса разработки, которые необходимо анализировать и контролировать, согласованно собираются и накапливаются в единой базе данных. Накапливаемые данные должны с необходимой точностью и детализацией отражать реальный процесс разработки. Процесс сбора данных должен гарантировать их полноту, корректность и согласованность. Опыт показывает, что около 90% проверок и больше при контроле процесса могут быть автоматизированы при должном покрытии процесса собираемыми данными.

Keywords: контроль процесса разработки; метрики; инструментальная панель; сбор данных о процессе.

1. Необходимость контроля процесса

Средние и крупные компании-разработчики ПО сталкиваются с необходимостью контроля своего процесса разработки, если они хотят

видеть этот процесс стабильным и предсказуемым, гарантирующим стабильный высокий уровень качества разрабатываемых продуктов. Также такой контроль необходим, если компании хотят усовершенствовать свой

процесс разработки. В этом случае контроль процесса позволит найти «узкие места» и оценить эффективность изменений.

Контроль процесса можно условно разбить на следующие части:

- Контроль соблюдения заданного процесса.
 - Проверка согласованности, полноты и корректности данных, собираемых различными системами поддержки разработки;
 - Проверка соблюдения стандартов процесса разработки, принятого в компании.
- Контроль эффективности и качества процесса.
 - Оценка качества разрабатываемых продуктов;
 - Оценка продуктивности разработки;
 - Оценка точности планирования;
 - Интервьюирование разработчиков и менеджеров относительно процесса.
- Принятие решений.
 - Принятие проектных решений по результатам проверок (в случае проектных аудитов);
 - Принятие решений о необходимости внесения изменений в процесс разработки.

Как правило, проверки и оценки при таком контроле проводятся в виде формальных процедур с использованием соответствующих контрольных таблиц (checklists). Результатом их является отчёт, содержащий следующую информацию:

- Список найденных замечаний и обобщённая оценка согласованности, полноты и корректности собираемых данных;
- Список найденных замечаний и обобщённая оценка соблюдения стандартов процесса разработки, сгруппированных по областям процесса;
- Список найденных замечаний по качеству разрабатываемых продуктов и обобщённая оценка их качества;
- Обобщённая оценка продуктивности разработки;
- Обобщённая оценка точности планирования.

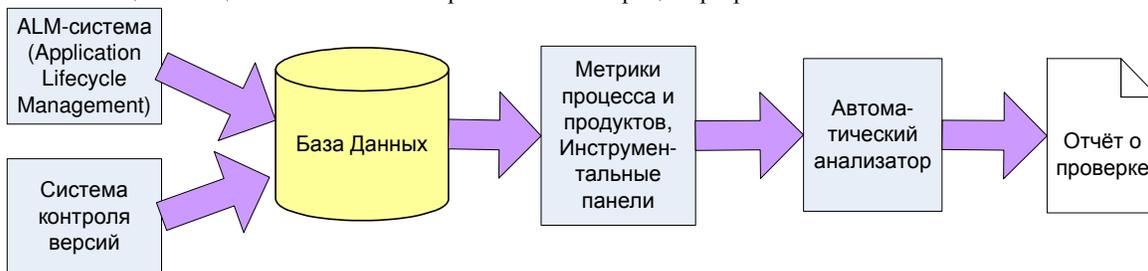


Рисунок 1. Примерная инфраструктура для автоматизации контроля процесса разработки.

Результаты таких отчётов обсуждаются на собраниях, делаются выводы о качестве и эффективности процесса, и принимаются решения о более тщательном контроле процесса (в случае его несоблюдения), о модернизации процесса (в случае обнаружения недостатков), о локальных изменениях процесса для конкретных ситуаций (если оказалось, что стандартный процесс не совсем удобен), о корректировке искажённых данных (если найдены замечания по корректности и согласованности данных).

Проведение вышеупомянутых проверок и подготовка отчётов может быть довольно длительной и дорогостоящей процедурой. Заметно упростить эту процедуру позволяет использование различных автоматизированных метрик процесса разработки, которые являются ключевым источником объективной информации о процессе. Более того, автоматизировать можно не только сбор необходимых метрик, но и сами такие проверки, что позволило бы существенно снизить стоимость контроля процесса разработки.

2. Инфраструктура для автоматизации контроля процесса

В первую очередь, инфраструктура, обеспечивающая возможность автоматизации контроля процесса, должна позволять согласованно собирать и накапливать данные обо всех значимых аспектах процесса разработки, которые необходимо анализировать и контролировать. Все эти данные могут собираться с помощью разных программных средств поддержки процесса разработки ПО, таких как ALM-системы (Application Lifecycle Management), системы bug-tracking, task/time-tracking, системы планирования, системы контроля версий кода и т.п. Кроме того, процесс в компании должен включать в себя обязательное согласованное отслеживание (tracking) и измерение всех значимых аспектов процесса разработки в этих программных средствах. Все собираемые данные должны с необходимой точностью и детализацией отражать реальный процесс разработки.

Говоря об измерениях процесса, следует назвать 3 базовых измеряемых величины процесса разработки ПО:

- Время (затрачиваемое на разработку);
- Размер (создаваемых артефактов);
- Количество дефектов (в артефактах).

На основе вышеперечисленных величин можно построить множество производных метрик для разных целей, которые будут использоваться при анализе качества процесса.

Инфраструктура для автоматизации контроля процесса должна приблизительно иметь вид, изображённый на рис. 1.

Центральным звеном такой инфраструктуры является единая база данных, содержащая в себе всю необходимую информацию о процессе разработки. Более-менее полную картину процесса можно получить, если в базе данных будет содержаться следующая информация:

- Иерархия продуктов и проектов с разбивкой на задачи.
- Иерархия департамента разработки ПО с данными обо всех разработчиках и менеджерах.
- Данные о времени, потраченном разными сотрудниками на ту или иную активность или задачу.
- Данные о созданных артефактах (требования, дизайн, тест-планы, код, проектная документация и т.п.) с подсчитанными размерами, причём должны храниться три значения для размера – добавленное, модифицированное, удалённое.
- Данные о найденных дефектах со всеми необходимыми атрибутами и историей изменений.
- Данные об инспекциях (в случае применения процесса инспекций), прочие данные, специфичные для процесса разработки, принятого в компании.
- Вспомогательные данные.

Наличие всех этих данных создаёт широкое пространство для анализа эффективности процесса разработки, поиска скрытых зависимостей, влияющих на качество и продуктивность разработки.

3. Корректность данных

Необходимым условием использования данных и вычисления метрик на их основе является корректность этих данных, иначе они будут бесполезными. Процесс сбора данных должен гарантировать их корректность. Метрики вычисляются на основе «сырых» данных, которые могут собираться как автоматически, так и вручную. На корректность (т.е. согласованность

и полноту) ручных данных влияет человеческий фактор, поэтому в них может таиться источник искажения метрик. Для решения этой проблемы можно использовать следующие приёмы:

- Возложение ответственности на исполнителей за корректность данных, относящихся к их работе (если допустил ввод некорректных данных, то обязан исправить).
- Периодические проверки согласованности данных и соблюдения стандартов независимыми экспертами, входящими в группу контроля процесса.

И всё равно даже при жёстком контроле корректности данных по разным причинам могут возникать локальные статистические выбросы, которые следует исключать при сборе статистических метрик.

4. Автоматизация контроля корректности данных

Итак, корректность данных необходимо регулярно контролировать, иначе упадёт их достоверность. В этом могут помочь автоматизированные средства. Можно выделить 3 способа автоматизации контроля корректности данных:

- Контроль полноты данных в момент ввода. Такой контроль производится непосредственно в самих информационных системах, в которые эти данные вводятся, в момент ввода данных, т.е. система просто не позволяет завершить операцию, если данные некорректны. Это самый эффективный способ. Главный его недостаток – его применимость сильно ограничена из-за ограниченности контекста отдельных операций ввода данных.
- Контроль с помощью автоматических уведомлений. Специальные программные агенты могут периодически или при наступлении определённых событий проверять корректность данных и при обнаружении несогласованности посылать специальные письма-уведомления людям, ответственным за эти данные.
- Периодические проверки с помощью автоматизированных средств при участии независимых экспертов (раз в неделю или несколько недель). Это более высокоуровневые проверки, которые целесообразно реализовывать с помощью вышеописанных способов. Результатом такой проверки будет список найденных замечаний, сгенерированный автоматически, и, возможно, расширенный замечаниями, найденными вручную экспертами.

Проверка полноты, согласованности и корректности данных является одной из задач контроля процесса разработки.

5. Процесс контроля и улучшения процесса

Процесс контроля и улучшения процесса разработки ПО (рис. 2) с использованием автоматизации состоит из следующих основных шагов:

- Проверка соблюдения процесса:
 - Автоматизированная проверка собранных данных на полноту, корректность и согласованность.
 - Исправление найденных проблем в данных. Возврат на предыдущий шаг, если были проблемы.
 - Автоматизированная проверка соблюдения стандартов процесса разработки.
- Проверка эффективности процесса:
 - Автоматизированная оценка качества разрабатываемых продуктов, продуктивности разработки, точности планирования;
- Обсуждение результатов проверок на собрании:
 - Просмотр и обсуждение найденных замечаний.
 - Обсуждение эффективности отдельных аспектов процесса разработки.
 - Принятие решений по улучшению процесса, а также проектных решений в случае проектных аудитов.
- Выполнение принятых решений.

6. Автоматизация проектных аудитов

Формальный процесс аудита проектов включает с себя проверку различных показателей проекта, выявление тенденций, проверку

соблюдения стандартов процессов и т.д. Набор всевозможных проверок составляет в контрольную таблицу (checklist), которая потом используется. Отдельные проверки в контрольной таблице следует описывать в формальном виде. Возможные шаблоны для таких проверок следующие:

- «должны быть введены такие-то данные» (проверка полноты данных или соблюдения стандартов процесса);
- «такие-то данные должны соответствовать тем-то» (проверка согласованности данных или соблюдения стандартов процесса);
- «параметр или метрика X в таких-то условиях должна находиться в пределах от Xmin до Xmax» - проверка на некий диапазон значений, который считается приемлемым и вычисляется на основе исторических данных (проверка соблюдения стандартов процесса или эффективности процесса).

Примеры проверок из практики компании

CQG:

- «Каждая активность типа «разработка требований», «проектирование», «разработка тест-плана», «кодирование» приводит к созданию артефактов соответствующего типа с задержкой не более 2-х недель» - пример проверки на полноту и согласованность данных.
- «Каждое изменение кода размером, превышающим 20 строк, проходит через формальный процесс инспекций» - пример проверки на соблюдение стандартов процесса.
- «Количество дефектов, найденных в ходе системного тестирования, меньше количества дефектов, найденных в ходе интеграционного тестирования, которое, в свою очередь, меньше количества дефектов, найденных в ходе инспекций» - пример проверки качества процесса.

Подобные проверки могут быть легко автоматизированы при наличии необходимой

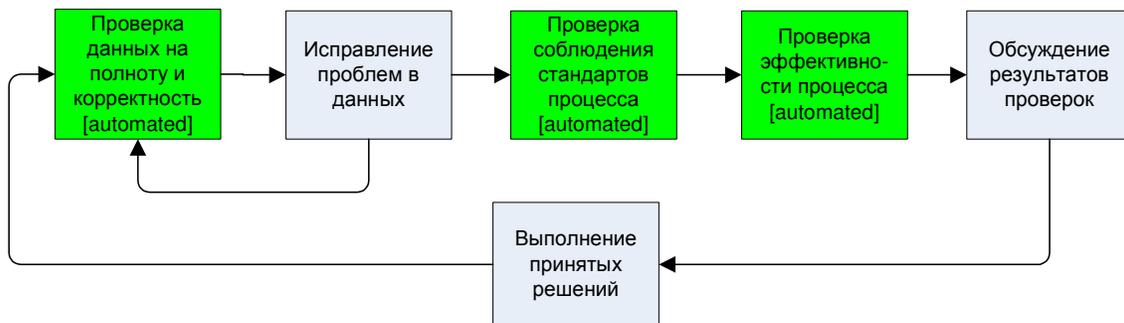


Рисунок 2. Процесс контроля и улучшения процесса разработки ПО. Автоматизированные шаги выделены подстрочкой [automated].

информации в базе данных.

Каждый проект имеет свою специфику, поэтому часто бывает, что стандартные проверки могут плохо подходить для конкретного проекта. Для таких случаев будет удобно наличие возможности подстройки параметров автоматизированных проверок в зависимости от особенностей проекта.

Чем больше проверок удастся автоматизировать, тем дешевле получится стоимость проведения аудита. Однако абсолютно всё автоматизировать не получится, поэтому без «ручного» экспертного анализа не обойтись, но пробелы в автоматизации следует максимально сокращать. Опыт показывает, что около 90% проверок при контроле процесса могут быть автоматизированы при должном покрытии процесса собираемыми данными.

7. Автоматизация контроля процесса разработки в компании CQG

Компания CQG является поставщиком данных и сервисов для биржевой торговли на основе разрабатываемых в компании программных систем. Штат департамента разработки насчитывает порядка 300 сотрудников, распределённых между 6 офисами в разных странах.

Разработчики ежедневно используют несколько средств автоматизации процесса разработки ПО, в которых параллельно с выполнением основных операций собираются данные о выполняемой работе:

- Система учёта рабочего времени и размеров произведённых артефактов с привязкой к конкретным проектам (собственная разработка).
 - Хранит иерархию департамента разработки ПО с данными обо всех разработчиках, менеджерах, и командах, а также иерархию продуктов и проектов.
 - В конце каждого рабочего дня каждый разработчик отмечает в этой системе, сколько времени на тот или иной проект и на ту или иную активность он потратил. Также разработчик вручную отмечает размер разработанных в этот день артефактов – требований, дизайна, тест-планов, документации.
 - Данные о размере разработанного кода попадают в эту систему автоматически. Также для подмножества проектов автоматически попадают данные о размерах требований и дизайна. Разработчику остаётся только привязать их к конкретному проекту.

- Система учёта отдельных задач и поддержки инспекций (собственная разработка).
 - В этой системе разработчики планируют и регистрируют свою работу на уровне отдельных небольших задач. Система полностью поддерживает методологию Personal Software Process (PSP).
 - Также эта система используется при проведении инспекций разрабатываемых артефактов. Сюда записываются данные о найденных в процессе инспекций замечаниях и потраченном на инспекции времени (с точностью до минут).
- Система bug-tracking и репозиторий требований (разработка на базе Siebel).
 - Сюда записываются данные о найденных дефектах с историей изменений и предложения по улучшению программных систем.
 - Также в эту систему заносятся все разработанные требования.
- Система контроля версий (CVS)
 - При внесении изменений в код количество добавленных, изменённых и удалённых вручную строк кода подсчитываются специальным скриптом и заносятся в первую систему. Автоматически сгенерированный, а также перенесённый из одной версии продукта в другую, код не учитывается при подсчёте.

Все эти системы частично интегрированы друг с другом. Раз в сутки все данные о процессе из этих систем импортируются в единую базу данных, которая используется для сбора метрик и анализа процесса. В будущем планируется внедрение единой интегрированной системы взамен четырёх вышеперечисленных.

Для целей анализа собранных данных о процессе в компании CQG были разработаны специальные программные средства (на платформе MS Excel), объединяющие в себе функциональность визуальных инструментальных панелей (dashboards) и автоматизированных анализаторов метрик. Разработано несколько типов таких средств, отличающихся областью применения: Team Dashboard, Project Dashboard, Team-Project Dashboard, System Dashboard, Maintenance Dashboard. Эти панели-анализаторы используют информацию из единой базы данных. Результатами работы таких панелей-анализаторов являются:

- Набор графических диаграмм, показывающих изменение во времени (с недельными периодами) основных показателей и метрик процесса.

- Набор текущих числовых значений основных метрик.
- Набор детализированных данных из БД, относящихся к выбранному подмножеству.
- Список замечаний, найденных при применении формальных контрольных таблиц (checklists).
- Обобщённая оценка соответствия процесса стандарту на основе автоматических проверок.

Можно выделить следующие основные метрики, отображаемые в инструментальных панелях:

- Метрики производительности:
 - Скорость кодирования (число строк кода в неделю, считая чистое время кодирования).
 - Производительность (число строк кода в неделю, считая всё время, затраченное на любые активности)
- Метрики качества:
 - Плотность дефектов разных типов (число дефектов на 1000 строк кода).
 - Объём переделок (в процентах от общего объема работ).
- Метрики формального процесса инспекций:
 - Скорость просмотра кода (число строк в час).
 - Плотность найденных замечаний (число замечаний на 1000 строк кода).
 - Процент дефектов от общего числа замечаний.
- Метрики точности планирования:
 - Отклонение реальных значений затрат от запланированных (в процентах).
- Себестоимость проекта (\$).
В компании CQG автоматизированный контроль процесса разработки применяется на нескольких уровнях:
 - Периодический аудит проектов (раз в 4-6 недель).
 - Периодический контроль стандартного процесса разработки в командах (раз в 2-3 недели).

- Периодический контроль процесса поддержки (maintenance) продуктов (в зависимости от частоты релизов).

Контроль проводится независимыми экспертами, которые совмещают роль разработчика (в других подразделениях) с ролью инженера по процессам разработки ПО. При контроле используются автоматизированные панели-анализаторы, упомянутые выше. Помимо замечаний, найденных автоматически, эксперты могут выявить вручную и добавить свои собственные замечания. Далее, в зависимости от категорий найденных замечаний, они либо сразу устраняются (например, некорректность или неполнота данных), либо обсуждаются на последующем собрании.

8. Заключение

Как известно, недостаточно задекларировать правила и стандарты процесса разработки. Чтобы он строго соблюдался и был эффективен, необходим его постоянный контроль. Автоматизация такого контроля возможна только в случае, если данные о процессе постоянно и согласованно собираются. Это в свою очередь накладывает дополнительные требования на процесс, обязывающие всех исполнителей аккуратно вводить данные о своей деятельности. По опыту CQG среднее суммарное время, которое тратит разработчик на работу со всеми программными инструментами сбора данных, составляет около 15 минут в день. Корректность и согласованность собираемых данных должна регулярно контролироваться. Но эти накладные расходы окупаются возможностью недорогого контроля соблюдения процесса разработки и анализа его эффективности, что, в свою очередь, приводит к более высокому и стабильному качеству разрабатываемых продуктов.