

ИНЖЕНЕРИЯ КАЧЕСТВА ПРОЦЕССОВ ПРОИЗВОДСТВА ПРОГРАММНЫХ СИСТЕМ С ПОМОЩЬЮ СЕТЕЙ ПЕТРИ

Л.Е. Матвеева

Институт кибернетики имени В.М. Глушкова НАН Украины,
Киев-03187, проспект Академика Глушкова, 40,
факс: 526 7418, тел: 526 0058,
e-mail: luda@iss.org.ua, lmatveeva@luxoft.com

Моделирование процессов производства программных продуктов и сервисов является важной частью их анализа и позволяет обнаруживать ошибки и слабые места на ранних стадиях жизненного цикла проектов, тем самым минимизируя затраты на их исправление. В данной работе применяется формализм сетей Петри для моделирования разработанной автором и внедренной в производство процедуры аудита качества с целью ее анализа, улучшения и настройки. Для анализа модели в виде ординарной сети Петри с одноцветными фишками решаются системы линейных однородных диофантовых уравнений над множеством натуральных чисел.

Analysis of software development processes models can contribute to the early detection of errors. By analyzing the models it is possible to prove specific properties of these models, and to gain deeper insights into their nature. In given paper the formal modelling technique of Petri nets is applied and is based on linear algebra methods of analysis in order to research some properties of auditing process developed by the author.

Введение

Мотивация. Начиная с середины восьмидесятых годов 20-го века и до настоящего времени в индустрии программного обеспечения продолжает нарастать интерес к применению различных моделей и стандартов процессов с целью обеспечения качества продуктов и сервисов. В частности, растущее количество организаций, внедряющих СММІ [1], привело в последние годы к созданию крупномасштабных программ по улучшению процессов разработки программного обеспечения. Положительные результаты внедрения таких программ достигаются, к сожалению, не всегда, и большие инвестиции не оправдываются. Вероятность успешного внедрения таких программ можно существенно увеличить, анализируя процессы и ключевые индикаторы их эффективности, поскольку они определяют успех или неудачу таких программ. Начальное описание процессов является лишь стартовой точкой работы по их внедрению. Только анализируя процессы в ходе реальной работы, регулярно определяя слабые места или возможность их появления, можно обнаружить системные проблемы и несоответствия и провести процедуру усовершенствования процессов.

Моделирование процессов. Моделирование процессов является важной частью их анализа и, соответственно, усовершенствования [2]. Модель процесса – это, как правило, структурированный набор элементов, который описывает характеристики эффективного процесса.

Известны различные способы моделирования процессов, например, с помощью построения: диаграмм состояний (state chart), диаграмм деятельности (activity diagram) или блок-схем (flow chart), диаграмм потоков данных (data-flow diagrams), а также Модели Показателей Процесса (Process Performance Model) и т. п. ([2–4]). Выбор типа модели зависит от целей, преследуемых в ходе усовершенствования процесса, задач, которые нужно решить и которые задают определенный аспект моделируемого процесса, например, требуется подвергнуть анализу, посредством моделирования последовательность выполняемых действий, ресурсы, данные или функции.

Модель Эффективности Процесса (Process Performance Model) включает набор базовых метрических данных атрибутов процессов (baselines) [2] и может использоваться для прогнозирования и анализа рабочих характеристик и эффективности стандартных процессов организации; оценивания прогресса в достижении целей по показателям проекта, а также для прогнозирования уровня возврата инвестиций в построение и усовершенствование процессов в организации.

Диаграмма деятельности обеспечивает представление процесса в виде определенного шаблона деятельности (т. е. набора активностей или последовательности действий), обеспеченного соответствующими ресурсами, с определенными ролями и ответственностями. Такая модель также может включать описание исключений, интерфейсов и рабочих продуктов, которые поставляются заказчику. Язык моделирования процесса должен обеспечить компактное представление модели, поскольку модели процессов разработки программного обеспечения обычно достаточно сложные. Требуется представлять основные объекты как атомарные единицы описания, что означает, например, рассматривать используемые в процессах инструментальные средства как черные ящики, описывающие их входное / выходное поведение, а не внутреннюю структуру.

Пример простой модели. Рассмотрим в качестве примера разработанную автором и внедренную в производство процедуру планирования и проведения аудитов качества [2] и построим ее модели разных типов.

Аудит качества – это объективная оценка соответствия процессов, продуктов и сервисов принятым в организации стандартам и процедурам. Данная деятельность является частью процесса Обеспечения Качества (Quality Assurance – QA), который в целом включает:

- определение целей по качеству и плана по управлению качеством в организации;
- отслеживание и сбор метрической информации и данных проектов;
- проведение аудитов качества (audits) в проектах.

Объективность оценки соответствия обеспечивается использованием критериев, относительно которых идет проверка индикаторов внедрения и институционализации производственных практик. Использование критериев означает наличие списков пунктов проверки, которое содержится в контрольных листах (checklists) аудитов, согласованных всеми участвующими в аудите сторонами.

Начнем с построения простой модели данной процедуры в виде одной из разновидностей диаграмм деятельности (activity diagram). Дадим необходимые определения для такой модели.

- Процесс (Process) – набор активностей, которые имеют последовательность во времени, входные и выходные условия и данные. Примеры процессов – анализ требований, проектирования архитектуры продукта, системное тестирование, проведение аудитов качества и т. п.
- Активность (Activity) имеет четкую постановку задания и, в целом, атомарный характер, является ответственностью одной персоны или группы. Примеры активностей процесса аудита качества – планирование аудита, проведение аудита, регистрация результатов, устранение несоответствий, выявленных в ходе аудита.
- Роль (Role) – это определенная зона ответственности. Примеры ролей: менеджер проекта (project manager), тестовый инженер (test engineer), инженер по обеспечению качества (QA engineer или quality engineer).
- Условие (Condition) – это или входное условие, которое должно выполняться перед началом определенного процесса или действия, или постусловие, которое будет выполняться по окончании данного процесса.

Могут также оказаться полезными (хотя и не используются в примере на рис.1) следующие элементы модели [2]:

- Поставка (Deliverable) — материальный выход/результат процесса.
- Исключение (Exception) – описание того, как изменить процесс, если происходит определенное ожидаемое или непредвиденное событие. Примеры исключений процесса: согласованное отклонение от стандартного процесса (Deviation Note), процедура временного замещения исполнителя определенной роли (substitution procedure).
- Взаимодействия (Communication) могут быть неформальные или формальные. Формальные взаимодействия – это, например, процедура утверждения поставки проектным менеджером; неформальные взаимодействия – это, например, обмен электронной почтой по поводу двусмысленности в документе.

С учетом данных определений модель процесса проведения аудита качества выглядит следующим образом (рис. 1).

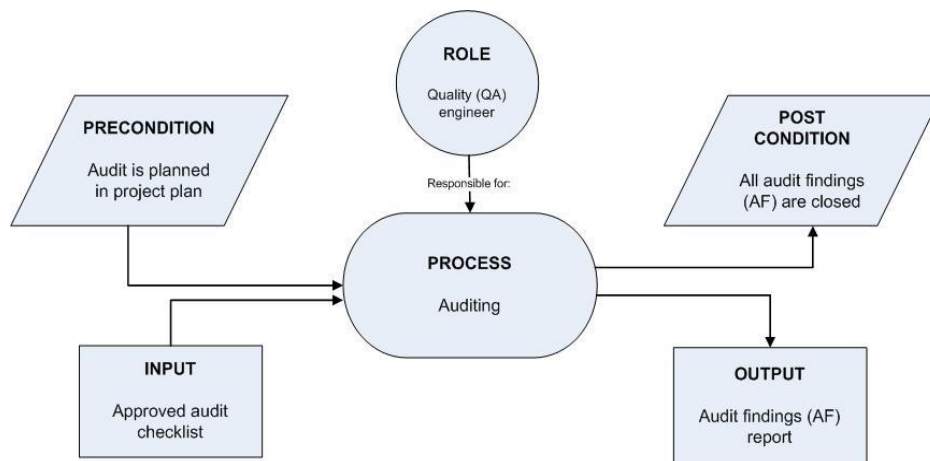


Рис. 1. Модель процесса аудита качества

Как видно из рис. 1, имеется элемент модели «Процесс», который представлен в виде овала. Предусловием является то, что аудит должен быть запланирован; постусловием модели является то, что все несоответствия, выявленные в ходе аудита, устранены; на входе в процесс имеется согласованный и настроенный для данного проекта контрольный лист (checklist) аудита; выходные данные процесса – это отчет о проведении аудита и исправлении обнаруженных несоответствий.

Проведем декомпозицию общего описания процесса на подпроцессы. В итоге данный процесс может быть представлен набором диаграмм последовательно выполняемых функций, состоящих из нескольких элементов «Активность», которые изображены в овалах (окантованных штриховой линией), на следующей схеме (см. рис. 2).

Для каждой активности в свою очередь могут быть представлены входные и выходные условия и данные. Однако, дальнейшая детализация этой модели не является целью данной работы.

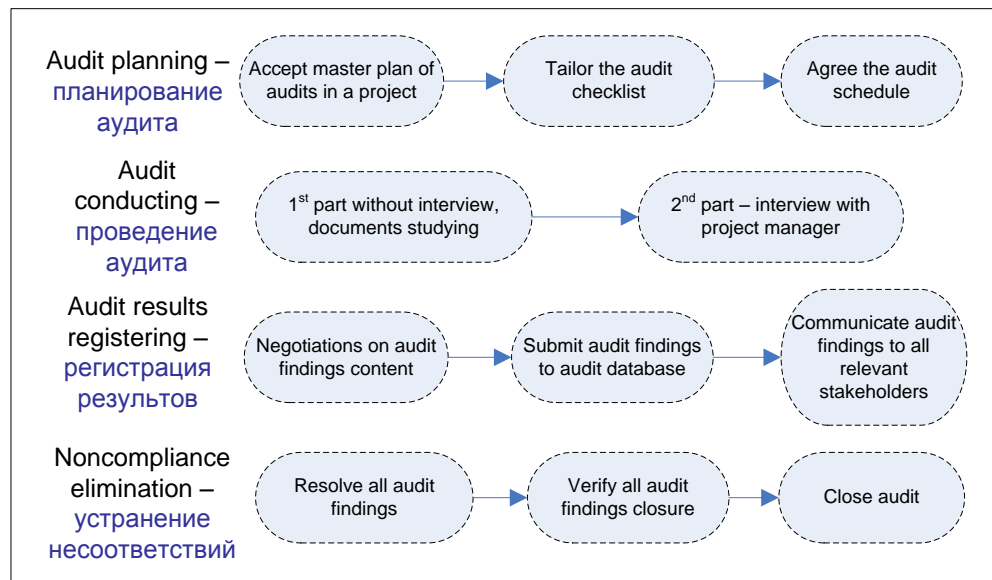


Рис. 2. Активности моделі процесу аудиту якості

Постановка задачі

В данній роботі пропонується метод моделювання з допомогою ординарних мереж Петри з одноцветними фішками [5, 6]. При цьому використовуються відомі визначення теорії мереж Петри [5 – 7] [5, 6]. Напамини лише, що Сеть Петри представляє собою двудольний орієнтований граф, що складається з вершин двох типів — місць і переходів, з'єднаних між собою дугами, вершини одного типу не можуть бути з'єднані безпосередньо. В місцях можуть розміщуватися фішки, здатні переміщатися по мережі. Событием називають спрацьовування переходу, при якому фішки з вхідних місць цього переходу переміщуються в вихідні місця. События відбуваються при виконанні деяких умов.

Дана математична модель є адекватним механізмом описання властивостей дискретних реактивних систем, які включають асинхронні елементи і розподілені частини. Відомо, що математичний апарат мереж Петри може застосовуватися для описання паралелізму моделюваної системи, планування розподілу ресурсів, координації різних цілей і т. д. [7, 8]. Для аналізу моделі в формі мережі Петри в данній роботі застосовуються методи лінійної алгебри [9, 10]. Розв'язуються системи лінійних однорідних діофантових рівнянь над множиною натуральних чисел. При цьому використовуємо відомі визначення – [7, 10, 11].

Інженерія (розробка) програмного забезпечення ([1], [12–16]) і теорія мереж Петри є дисциплінами різної природи, хоча інженерія програмного забезпечення має тісні зв'язки зі своєю базовою дисципліною – інформатикою. Дослідження в області розробки програмного забезпечення орієнтовані на пошук рішень, які покривають різні аспекти промислового виробництва складних великих програмних систем. В той час як дослідження в області мереж Петри ґрунтуються на вивченні додатків і властивостей спеціальної математичної моделі.

Однак, моделювання і аналіз систем з допомогою мереж Петри може застосовуватися на різних фазах процесу розробки програмного забезпечення. Відомою областю застосування формалізму мереж Петри є формальна специфікація і верифікація вимог і дизайну програмної системи ([10, 11, 17]), [8]. Покажемо, що і інші задачі інженерії програмного забезпечення можуть розв'язуватися з допомогою застосування формалізму мереж Петри. В данній роботі наведено приклад того, як можна застосувати мережі Петри для моделювання одного з процесів виробництва програмного забезпечення з метою його аналізу, покращення і налаштування, а саме: процесу аудиту якості. Аналіз моделей процесів в формі мереж Петри може сприяти ранньому виявленню помилок в процесах. Це дуже важливо, так як якість продукту в значній мірі визначається якістю процесу виробництва цього продукту; і тому в результаті аналізу процесів розробки і наступного удосконалення, яке орієнтовано на якість продукту, покращується якість продукту. Аналізуючи такі моделі, можна довести їх специфічні властивості і отримати більш глибокі знання про їх природу, а значить виявити можливі області для покращення цих процесів, які вони моделюють. Аналіз моделі дозволяє покращувати управління процесами.

Моделювання процесу мережею Петри і її аналіз

На рис. 3 представлено ще одна більш детальна, ніж во Введенні, модель процесу аудиту якості. Формат данньої діаграми відповідає вимогам стандарту ISO/TS 16949:2002. Ролі процедури аудиту (горизонтальні смуги на схемі): Інженер з Забезпечення Якістю (QA Engineer), Керівник Проекту (Project Manager або PM). AF (audit finding) – це невідповідність, виявлена в ході аудиту.

Основные шаги процедуры, представленной на рис. 3.

1. Инициация проведения аудита качества состоит в обеспечении предварительных условий его проведения и входных данных: подразумевается наличие плана аудитов проекта и предварительно согласованных контрольных листов (checklists) аудитов. Всякий контрольный лист адаптируется к нуждам конкретного проекта и утверждается Руководителем Проекта. Настройка аудита и его контрольного листа производится с целью учесть специфику проекта и проводить аудирование более целенаправленно.
2. Согласовывается план-график аудита.
3. Аудит проводится.
4. Данные аудита регистрируются. Заполняется отчет о проведении аудита, который рассылается всем заинтересованным сторонам, после соответствующего согласования с указанием обнаруженных несоответствий.
5. Обеспечивается исправление несоответствий, обнаруженных в ходе аудита.
6. Инженер по Обеспечению Качества производит проверку устраненных несоответствий.
7. Если все зарегистрированные несоответствия устранены, аудит считается завершенным.

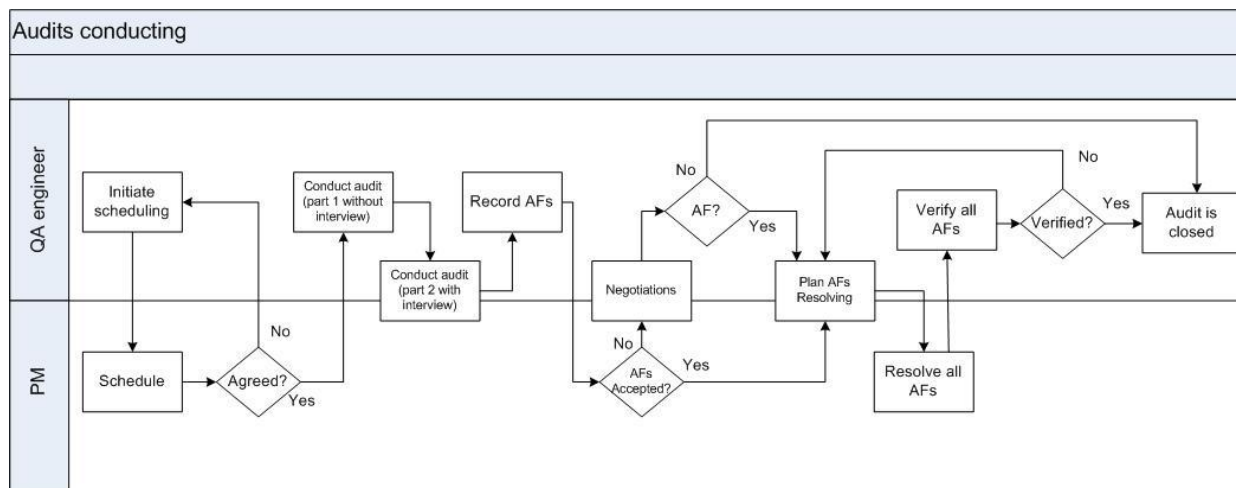


Рис. 3. Схема проведения аудита качества

Теперь представим модель данной процедуры на рис. 4, используя формализм ординарных сетей Петри с одноцветными фишками ([5 – 7]), а затем проанализируем ее свойства, используя определения и метод, представленный в [9–11, 18] и [19].

Места сети (рис. 4) интерпретируем следующим образом: P1 – условие инициации проведения аудита (см. первый шаг процедуры в начале данного раздела); P2 – условие достижения договоренности о дате проведения аудита; P3 – условие выделения Руководителем Проекта необходимых ресурсов; P4 – условие наличия графика проведения аудита; P5 – проведение первой части аудита – изучение проектных документов; P6 – проведение второй части аудита – интервью с Руководителем Проекта, регистрация обнаруженных несоответствий; P7 – согласование обнаруженных несоответствий с Руководителем Проекта; P8 – Инженер по Обеспечению Качества уточняет формулировку обнаруженных несоответствий; P9 – согласование уточненной формулировки обнаруженных несоответствий; P10 – дополнительные аргументы по выявленным несоответствиям; P11 – условие наличия графика устранения выявленных несоответствий; P12 – условие готовности к проверке Инженером по Обеспечению Качества устраненных несоответствий; P13 – имеются аргументы по недостаточным действиям для устранения выявленных несоответствий; P14 – условие принятия решения по результатам проверки устраненных несоответствий; P15 – условие закрытия аудита.

Переходы сети (рис. 4) интерпретируем следующим образом: t1 – событие инициации проведения аудита; t2 – заявлено несогласие с предварительными условиями его проведения или входными данными; t3 – достигнуто согласие с предварительными условиями проведения аудита и входными данными; t4 – проведение первой части аудита (изучение проектных документов); t5 – проведение второй части аудита (интервью); t6 – регистрация обнаруженных несоответствий; t7 – несогласие с формулировкой несоответствий; t8 – планирование работ по устранению несоответствий; t9 – отклонить несоответствие, как такое, которое не является нарушением процесса, и согласовать это всеми сторонами; t10 – планирование работ по устранению несоответствий; t11 – событие устранения несоответствий; t12 – событие проверки устранения несоответствий; t13 – событие перепланирования работ по устранению несоответствий; t14 – событие закрытия несоответствий; t15 – событие завершения аудита качества.

Возможна начальная разметка сети: $M_0 = (1,0,1,0,0,0,0,1,0,1,0,0,1,0,0)$, интерпретация которой следующая: проведение аудита инициировано – контрольный лист согласован, аудит в проекте запланирован, то есть выделены все необходимые ресурсы (Руководитель Проекта выделил время и обеспечил инструментальные средства для автоматизации процесса); в проекте имеется выделенный Инженер по Обеспечению Качества, который готов проводить аудит и согласование с Руководителем Проекта, давать необходимые разъяснения, а также производить проверку того, что все выявленные в ходе аудита несоответствия действительно устранены.

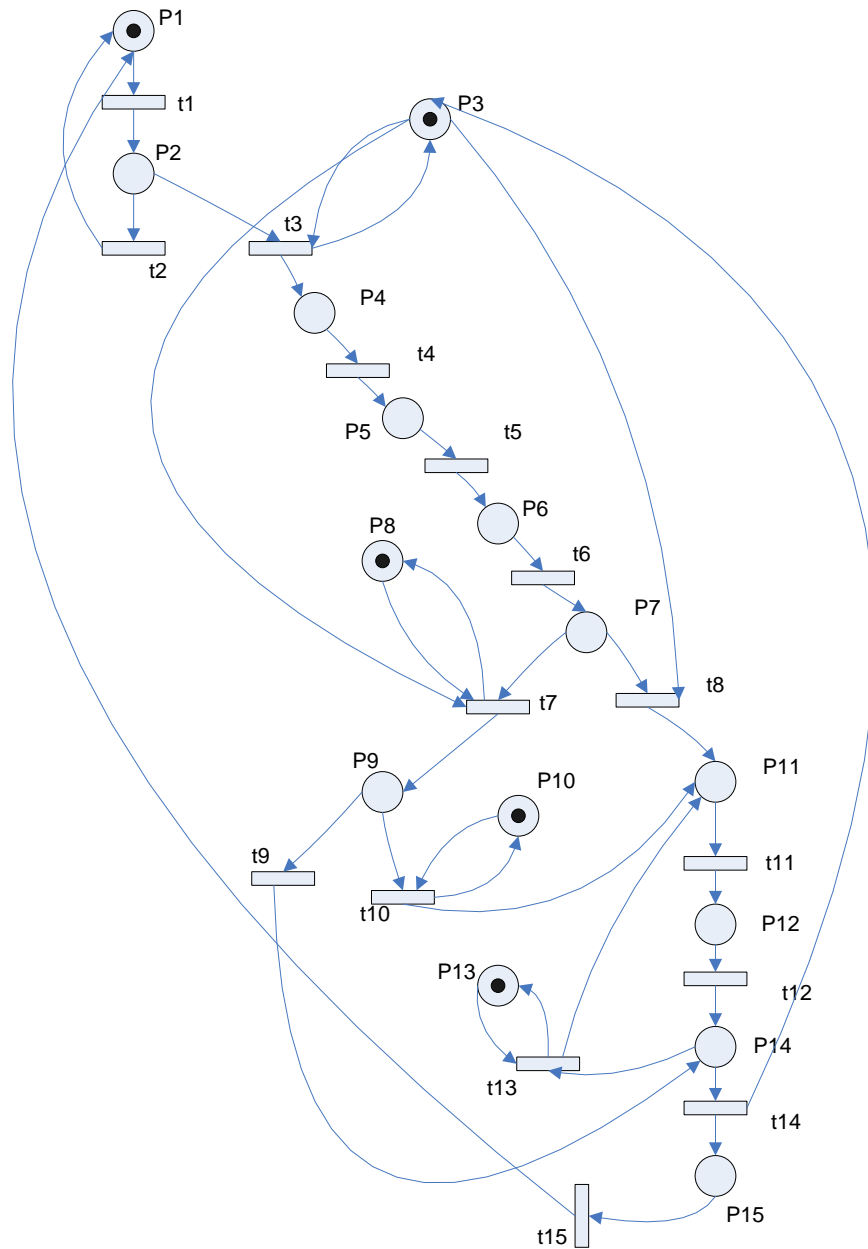


Рис. 4. Сеть Петри моделюющая процедуру проведения аудита качества

Для анализа свойств данной модели строим по сети (рис. 4) ее матрицу инцидентности A :

	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12	t13	t14	t15
P1	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	1
P2	1	-1	-1	0	0	0	0	0	0	0	0	0	0	0	0
P3	0	0	0	0	0	0	-1	-1	0	0	0	0	0	1	0
P4	0	0	1	-1	0	0	0	0	0	0	0	0	0	0	0
P5	0	0	0	1	-1	0	0	0	0	0	0	0	0	0	0
P6	0	0	0	0	1	-1	0	0	0	0	0	0	0	0	0
P7	0	0	0	0	0	1	-1	-1	0	0	0	0	0	0	0
P8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P9	0	0	0	0	0	0	1	0	-1	-1	0	0	0	0	0
P10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P11	0	0	0	0	0	0	0	1	0	1	-1	0	1	0	0
P12	0	0	0	0	0	0	0	0	0	0	1	-1	0	0	0
P13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P14	0	0	0	0	0	0	0	0	1	0	0	1	-1	-1	0
P15	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1

Вычисляем минимальное порождающее множество T-инвариантов, решая однородную систему уравнений $Ax = 0$, где A – матрица инцидентности верифицируемой сети Петри; данное множество представлено 5 векторами, которые соответствуют последовательностям срабатывания переходов:

$$\begin{aligned} T1 &= (1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) \\ T2 &= (1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1) \\ T3 &= (0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0) \\ T4 &= (1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1) \\ T5 &= (1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1) \end{aligned}$$

Затем решим однородную систему уравнений $A^T y = 0$, где A^T – транспонированная матрица инцидентности верифицируемой сети Петри. Вычисляем минимальное порождающее множество S-инвариантов, оно представлено 5 векторами:

$$\begin{aligned} S1 &= (0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0) \\ S2 &= (0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0) \\ S3 &= (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0) \\ S4 &= (0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0) \\ S5 &= (1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1) \end{aligned}$$

Результаты анализа данных инвариантов.

- Из минимального порождающего множества T-инвариантов следует, что все переходы покрыты T-инвариантами, и, значит, сеть обладает структурным свойством L3-живости.
- Сеть является повторяемой, то есть любая последовательность срабатываний переходов может повториться теоретически бесконечное число раз.
- Сеть является непротиворечивой, так как каждый переход встречается хотя бы один раз при переходе от начальной разметки к ней же.
- Из минимального порождающего множества S-инвариантов следует, что все места покрыты S-инвариантами, что говорит о том, что все места данной СП ограничены, и, следовательно,
- Данная СП структурно ограничена.

Из анализа модели можно сделать вывод, что система, представленная данной моделью, не содержит дедлоков и ограничена по ресурсам.

Усовершенствование процесса в результате анализа модели. Данная сеть и результаты ее анализа могут использоваться для совершенствования процедуры аудита качества, представленной на рис. 1, 2 и 3. При создании модели процедуры в виде сети Петри выяснилось, что Руководителя Проекта должен запланировать время для работы над результатами аудита (анализ и устранение несоответствий). Действительно, связь между переходами T7, T8 и T3 и местом P3 моделирует распределение ресурса Руководителя Проекта. Инженер по Обеспечению Качества обязан это учесть при описании данной процедуры. Схема (рис. 3) и описание должны быть соответствующим образом отредактированы, в процедуру (представленную в начале данного раздела) должен быть добавлен шаг, описывающий выделение времени для обработки выявленных в ходе аудита и согласованных с Инженером по Обеспечению Качества несоответствий.

Для полноты представления темы моделирования процессов нужно указать на риски такого внедрения процессов в организации, которое основано на их моделировании. Эти риски связаны с тем, что, во-первых, при внедрении модели существует опасность слишком больших расходов, которые не являются обоснованными бизнес-целями организации; во-вторых, все модели являются неполными; в-третьих, для корректного использования модели с пользой для производства нужно глубокое профессиональное ее понимание и оценка.

Заключение и выводы

Инженер Группы Обеспечения Качества обнаруживает возможности совершенствования процессов, занимаясь анализом процессов, поиском типов образцов низкой эффективности, дефектов или других проблем [2]. Моделирование процессов является важной частью их анализа. Модель является важной потому, что обеспечивает отправную точку анализа, общий язык и виденье, а также использование опыта всех участвующих в создании модели процесса. Кроме того модель процесса дает метод оценивания и проведения диагностики состояния текущих процессных практик; помогает обнаруживать ошибки и слабые места на ранних стадиях жизненного цикла проектов, тем самым минимизируя затраты на их исправление; помогает устанавливать цели и приоритеты по совершенствованию процессов.

В данной работе рассмотрены различные модели процедуры проведения аудита качества, в том числе построена модель с использованием формализма ординарных сетей Петри с одноцветными фишками ([5,6]). Проводится анализ свойств сети с применением методов линейной алгебры [7, 9, 10]. Приведен пример того, как результаты моделирования могут использоваться для совершенствования данной процедуры.

1. CMMI for Development, Version 1.2 // <http://www.sei.cmu.edu/pub/documents/06.reports/pdf/06tr008.pdf>
2. *Матвеева Л.С., Волков В. А.* Процесс разработки программного обеспечения. Від теорії до практики. – Київ. – 2008.
3. *DeMarco T.* Structured Analysis and Systems Specification // Englewood Cliffs, N.J.: Prentice-Hall, 1979. – 352 p.
4. *Yourdon E.* Just Enough Structured Analysis // 2006. – <http://www.yourdon.com/jesa/jesa.php>
5. *Peterson J.L.* Petri Net Theory and the Modelling of Systems // Prentice-Hall, Englewood Cliffs, New Jersey, (April 1981). – 290 p.
6. *Котов В. Е.* Сети Петри // М.:Наука, 1984. – 157 с.
7. *Murata T.* Petri Nets: Properties, Analysis and Applications // In Proc. of the IEEE. – 1989. –Vol. 77. – N 4. – P. 541–580.
8. *E.M. Clark, J.M. Wing.* Formal methods: State of the Art and Future Directions // ACM Computing Surveys. – Vol. 28. – N 4. –1996. – P. 626-643.
9. *Матвеева Л.Е.* Анализ сетей Петри с помощью инвариантов // Искусственный интеллект. – 2001. – № 3. – С. 243–250.
10. *Крывий С.Л., Матвеева Л.Е.* Формальные методы анализа свойств систем // Кибернетика и системный анализ.–2003. – № 2. – С. 15–36.
11. *Матвеева Л.Е.* Автоматическая система анализа и верификации телекоммуникационной системы, описанной на языке MSC, с помощью формализма сетей Петри // Проблемы программирования. – 2004. – № 2–3. – С. 108–117.
12. *Е.М. Лаврищева.* Программная инженерия – научная и инженерная дисциплина // Кибернетика и системный анализ. – 2008. – № 3. – С. 19 – 28.
13. *Е.М. Лаврищева.* Классификация дисциплин программной инженерии. – Кибернетика и системный анализ. – 2008. – № 6. – С. 5 – 14.
14. *Ф.И. Андон, Г.И. Коваль, Т.М. Коротун, Е.М. Лаврищева, В.Ю. Сулов.* Основы инженерии качества программных систем // Второе издание. НАНУ Институт программных систем. Издательский дом «Академперіодика», Киев – 2007.
15. *W.S. Humphrey.* Managing the software process // SEI series in software engineering, Addison-Wesley. – 1989.
16. *J.M. Juran, M. G. Frank, Jr.* Quality Planning and Analysis // 2nd ed., New York: McGraw-Hill. – 1980.
17. *Gold R.* Petri Nets in Software Engineering // Arbeitsberichte - Working Papers, Ingolstadt. – 2004.
18. *Крывуу S., Матвеева Л., Лопатина М.* Automatic Modeling and Analysis of MSC-specified Systems // Fundamenta Informaticae, Vol.67, N. 1 – 3, August 2005. – P. 107–120.
19. *Крывий С.Л., Матвеева Л.Е., Лопатина М.В.* Алгоритм Построения TSS-множества для СЛОДУ в натуральных числах // «Алгебра, логика и кибернетика» (АЛК-2004). – Матер. Междунар. конф., посвященной 75-летию со дня рождения А.И. Кокорина, Россия, Иркутск, 25–28 августа, 2004. – С. 169–171.