

# COMPUTER-AIDED ENGINEERING

## СИСТЕМЫ АВТОМАТИЗАЦИИ ПРОЕКТИРОВАНИЯ

УДК 004.896:004.94

Н.Г. Ярушкина, И.А. Тимина

### МОДЕЛЬ И СРЕДСТВА УПРАВЛЕНИЯ ПРОЕКТИРОВАНИЕМ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ НА ОСНОВЕ ДИНАМИКИ МЕТРИК ПРОГРАММНОГО КОДА<sup>1</sup>

**Ярушкина Надежда Глебовна**, доктор технических наук, профессор, окончила радиотехнический факультет Ульяновского государственного технического университета. Первый проректор - проректор по научной работе Ульяновского государственного технического университета. Имеет более 250 работ в области мягких вычислений, нечеткой логики, гибридных систем. [e-mail:jng@ulstu.ru].

**Тимина Ирина Александровна**, окончила факультет информационных систем и технологий УлГТУ по специальности «Прикладная информатика (в экономике)». Ассистент кафедры «Информационные системы» Ульяновского государственного технического университета. Имеет статьи и научные труды в области интеллектуального анализа данных. [e-mail: timina\_i@mail.ru].

#### Аннотация

В статье рассматривается вопрос управления проектами, связанными с разработкой продуктов программного обеспечения, посредством использования автоматизированной системы контроля версий (СКВ), и анализом метрик программного кода. Для решения поставленной задачи предлагается исследование работы СКВ с последующим использованием компоненты анализа данных ведения проекта на основе применения модели временного ряда (ВР), построения ВР нечетких тенденций, кластеризации для выделения доминирующей нечеткой тенденции, выделения ВР-предиката, который имеет влияние на исследуемый ВР, степени сходства между ВР, их корреляции, прогнозирования и корректировки прогноза на основе зависимости метрик событий в процессе разработки проекта. В качестве метрик программного кода были использованы ВР числа ошибок из общего числа изменений, количество улучшений также из числа изменений, количество введенных новых функций. Для прогнозирования была выбрана гипотеза сохранения тенденции. Предложенный подход исследован в примерах.

Ключевые слова: система контроля версий, временной ряд, нечеткая тенденция, прогнозирование, корректировка прогноза.

### AUTOMATED SYSTEM MODEL AND CONTROL TOOLS ON THE BASE OF PROGRAM CODE METRICS HISTORY

**Nadezhda Glebovna Yarushkina**, Doctor of Engineering, Professor, First Vice-Rector – Vice-Rector for Science of Ulyanovsk State Technical University; graduated from the Faculty of Radioengineering of Ulyanovsk State Technical University ; an author of more than 250 papers in the field of soft computing, fuzzy logic, and hybrid systems. e-mail: jng@ulstu.ru.

<sup>1</sup> Работа выполнена при поддержке РФФИ, проект № 14-01-31092.

**Irina Aleksandrovna Timina**, Assistant at the Department of Information Systems at Ulyanovsk State Technical University; graduated from the Faculty of Information Systems and Technologies of USTU with a specialty of Applied Informatics (in Economics); an author of articles in the field of data mining. e-mail: timina\_i@mail.ru @ulstu.ru.

#### Abstract

The article discusses the issue of project management associated with the development of software products through using automated version control system (VCS) and the analysis of program code metrics. This problem is solved through studying VCS functioning with the further use of the data analysis component of the project management based on the application of the time series (TS) model, the construction of fuzzy TS trends, clustering for dominant fuzzy trends separation, extracting time series predicate, the similarity measure of time series, their correlation, prediction and correction of the forecast. Time series of the number of errors in the total number of changes, the number of improvements in the same number of changes, the number of new functions were used as program code metrics. The hypothesis of the trend permanency was chosen for prediction. The given approach was examined on the examples.

Keywords: version control system, time series, fuzzy trend, forecasting, forecast adjustment.

#### ВВЕДЕНИЕ

В настоящее время при разработке достаточно крупных и сложных проектов, непосредственно связанных с созданием продуктов программного обеспечения, для хранения исходных кодов разрабатываемой программы наиболее широко используется система управления версий (от англ. *Version Control System*, *VCS* или *Revision Control System*) разрабатываемого проекта. Эта система представляет собой программное обеспечение для облегчения работы с изменяющейся информацией, позволяет хранить несколько версий одного и того же документа и при необходимости возвращаться к более ранним версиям, а также определять, кто и когда сделал то или иное изменение, и многое другое.

В связи с этим основным инструментом управления проектированием автоматизированной системы (АС) и является система контроля версий. Существуют различные виды данных систем [1]:

- CVS (*Concurrent Versions System*). Система поддерживает версиюность файлов. Это означает, что любой файл, появляющийся в проекте, получает собственный номер версии (обычно — номер 1, условной «нулевой» версией файла считается пустой файл с тем же именем). При каждой фиксации разработчиком изменений, затрагивающих файл, соответствующая часть фиксируемых изменений применяется к файлу, и файл получает новый, обычно следующий по порядку, номер версии. Поскольку фиксации обычно затрагивают только часть файлов в репозитории, номера версий файлов, имеющиеся на один и тот же момент времени, со временем расходятся, и проект в целом (то есть весь набор файлов репозитория) фактически никакого «номера версии» не имеет, так как состоит из множества файлов с различными номерами версий.

- SubVersion (*SVN*). Для других систем понятие «версия» относится не к отдельному файлу, а к репозиторию целиком. Вновь созданный пустой репозиторий имеет версию 1 или 0, любая фиксация изменений приводит к

увеличению этого номера (то есть даже при изменении одного файла на один байт весь репозиторий считается измененным и получает новый номер версии). Номера версии отдельного файла здесь фактически не существует, условно можно считать таковым текущий номер версии репозитория (то есть считать, что при каждом изменении, внесенном в репозиторий, все его файлы меняют номер версии, даже те, которые не менялись). Говоря о «версии файла» в таких системах, имеют в виду ту версию репозитория, в которой файл был последний раз (до интересующего нас момента) изменен.

- GIT – open-source инструмент, контентно-адресуемая файловая система, организующая поддержку нелинейной разработки (тысячи параллельных веток) одновременно несколько больших проектов. Сравнивая GIT с «родственными» ей распределенными системами, можно отметить, что GIT изначально идеологически ориентирована на работу с изменениями, а не с файлами, «единицей обработки» для нее является набор изменений или патч. Эта особенность прослеживается как в структуре самой системы (в частности, в структуре репозитория), так и в принципах построения команд; она отражается на производительности системы в различных вариантах ее использования и на достоинствах и недостатках GIT по сравнению с другими DVCS.

- Mercurial. Кроссплатформенная распределенная система управления версиями, разработанная для эффективной работы с очень большими репозиториями кода. Наряду с традиционными возможностями систем контроля версий, Mercurial поддерживает полностью децентрализованную работу (отсутствует понятие основного хранилища кода), ветвление (возможно вести несколько веток одного проекта и копировать изменения между ветками), слияние репозитория (чем и достигается «распределенность» работы). Поддерживается обмен данными между репозиториями через HTTP/HTTPS, SSH и вручную при помощи упакованных наборов изменений.

- ClearCase. Особенностью данной системы является поддержка сохранения новой и последней версий: большинство версий сохраняется в виде дельт, но периодически (по специальной команде администратора) выполняется сохранение версий всех файлов в полном виде; такой подход обеспечивает максимально полное восстановление истории в случае повреждения репозитория.

Перед работой разработчик соединяется с сервером и копирует на компьютер проект в виде дерева каталогов. В большинстве случаев на локальный диск помимо основной копии копируется дубликат рабочей версии проекта. При работе с проектом разработчик вносит изменения только в основную рабочую копию. Вторая локальная копия остается в качестве образца. С помощью нее можно, без обращения к серверу, выяснить какие изменения были внесены в интересующий файл или в проект в целом.

С помощью ресурса GitHub, который является «надстройкой» GIT, отображается статистика реализации проекта с его изменениями, релизами, сведениями о разработчиках и т.п. [2].

Разработчик трансформирует проект, внося изменения во входящие файлы проекта его рабочей копии, которые соответствуют проектному заданию. Данная работа выполняется локально и не требует обращений к серверу VCS.

После каждого этапа работы над заданием разработчик записывает свои изменения (**commit**) на сервер. Commit в VCS представляются данными с именем сотрудника, датой внесения изменений в проект, текстом соответствующего вносимого изменения, информацией о ветке, на которой они были внесены. Также можно просмотреть, кем были внесены commit, после которых был создан релиз проекта.

При выполнении объемных работ рабочими группами не совсем удобно выполнять мелкие изменения в проекте посредством правки рабочей копии и дальнейшего изменения прямо в главной ветви на сервере. В связи с этим создаются ветви (**branch**) (то есть «отхождение» от главной ветви) в каком-то варианте нового проекта, работа в котором проходит параллельно с изменениями основной версии. Проект может иметь несколько ветвей, параллельных работ, каждая из которых может иметь статус «Активная», «Закрытая», «Объединенная» и т. п. В каждой ветви существует определенное количество точек из основной версии и точек «наперед», то есть возможных вариантов кода, которые можно проверить либо откатить назад, либо сделать основной веткой [3].

Новый выпуск проекта (**release**) содержит номер версии, информацию о разработчике, который его создал, комментарий о том, что данный релиз имеет нового и исправленного, кем и какие commit были внесены, сколько всего было внесено изменений относительно последнего релиза.

В VCS регистрируется множество изменений, релизов, ошибок, линий, ветвлений и т. д., но средств анализа метрик, которые позволят управлять эффективностью проекта, – нет.

## 1 Постановка задачи. Основания сущности модели

Чтобы построить средство управления проектом необходимо:

- отобразить метрики, по математической природе;
  - рассмотреть зависимости;
  - построить формальную модель;
  - построить средство управления проектами.
- Метриками будут являться свойства сущностей. К сущностям разрабатываемого проекта относятся:
- commit (рядовой разработчик, дата);
  - release (множество разработчиков, мастер, дата);
  - ветка;
  - ошибка (bug);
  - улучшение (improvement);
  - новая функциональная возможность (New Feature).

Улучшениями при разработке проекта считаются изменение какой-либо его части, повышение качества, усовершенствование того или иного подхода в системе.

Новые функциональные возможности – это запланированные действия программистов для выпуска очередного релиза.

На основе выделенных сущностей за определенный период времени могут быть отобраны следующие метрики:

- количество изменений;
- количество стабильных выпусков проекта;
- число ошибок из общего числа изменений;
- количество улучшений так же из числа commit;
- количество введенных новых функций.

В процессе реализации проекта от момента утверждения стабильной версии до выпуска нового релиза ставится определенное количество задач по введению новых функций системы, исправлению соответствующих ошибок по вводу данных функций и добавлению улучшений (исправленных ошибок). Введение новых функций оказывает влияние на добавление улучшений. Данные новых функций являются предикатом, то есть имеются некоторое подобие и зависимость между указанными метриками. Аналогичная взаимосвязь есть между появлением ошибок и улучшениями. Появившиеся ошибки являются предикатом добавленных улучшений.

В зависимости от миссии компании или конкретного проекта перед работниками могут стоять различные первостепенные задачи – это либо подчистка ошибок, либо развитие проекта за счет совершенствования.

Одной из задач управления проектом является задача прогнозирования. Прогноз с учетом основных задач ведения проекта (например, прогноз получения ошибок) покажет успешность проекта, ориентир на стабильную версию и позволит руководителю верно распределять задания сотрудникам.

## 2 ФОРМАЛИЗАЦИЯ ПОСТАВЛЕННОЙ ЗАДАЧИ

Модель анализа и управления проектами можно представить следующим образом

$$\{C_t, R_t, B_t, I_t, F_t, R^{BI}, R^{IF}\},$$

где  $C_t$  – временной ряд commit,

$R_t$  – временной ряд release,

$B_t$  – временной ряд bug,

$I_t$  – временной ряд improvement,

$F_t$  – временной ряд New Feature,

$R^{BI}$  – зависимость bug от improvement,

$R^{IF}$  – зависимость improvement от New Feature.

Данные проекта могут быть представлены в виде дискретных ВР. Чтобы построить модель ВР для целей решения данной задачи, предлагается преобразование исходного дискретного ВР  $Y = \{t_i, x_i\}, i \in [1, n]$ , где  $t_i$  – некоторый момент времени,  $x_i$  – уровень ВР, в нечеткий временной ряд (НВР)  $\tilde{Y} = \{t_i, \tilde{x}_i\}, i \in [1, n]$ ,  $X = \{x_i\}, \tilde{x}_i \in \tilde{X}$ , где  $\tilde{x}_i$  – нечеткая метка. Затем НВР преобразуется в ВР нечетких элементарных тенденций (НЭТ), который обозначим  $\tau_i = ETend(\tilde{x}_i, \tilde{x}_{i+1})$ . Под  $ETend$  понимается функционал, который реализуется на основе операций:  $TTend$  – определение типа тенденции и  $RTend$  – определение интенсивности тенденции по специальной лингвистической шкале, построенной для исходного ВР. В результате описанных преобразований получается несколько ВР, четких числовых и нечетких лингвистических, например, ВР параметров нечетких тенденций [4–7].

Нечеткая тенденция (НТ) НВР – нечеткая метка, выражающая характер изменения (систематическое движение) последовательности нечетких уровней НВР в заданном интервале времени. Нечеткая тенденция выражает поведение НВР в лингвистическом виде, например: «Рост», «Падение», «Стабилизация», «Колебания», «Хаос» [8].

Обобщенную модель НВР НТ представим:

$$\tau_i = f(\tau_{i-1}, \dots, \tau_{i-d}), \quad (1)$$

где  $d$  – фиксированное число, параметр модели;

$\tau_i, \tau_{i-1}, \dots, \tau_{i-d}$  – последовательность НТ;

$f$  – некоторая нечеткая зависимость.

Каждая НТ  $\tau \in \mathfrak{T}$  НВР  $\tilde{Y} = \tilde{x}_i, t = 1, 2, \dots$  может быть представлена структурной моделью в виде кортежа отношения, построенного на декартовом произведении свойств НТ  $\tilde{V} \times \tilde{A} \times \Delta T \rightarrow \mathfrak{T}$ :

$$\tau = \langle \tilde{v}, \tilde{a}, \Delta t, \mu \rangle, \quad (2)$$

где  $\tau$  – наименование НТ из множества  $\mathfrak{T}, \tau \in \mathfrak{T}$ ;

$\tilde{v}$  – тип НТ,  $\tilde{v} \in \tilde{V}, \{\text{Падение, Рост, Стабильность}\}$ ;

$\tilde{a}$  – интенсивность НТ,  $\tilde{a} \in \tilde{A}, \{\text{Сильное, Среднее, Слабое}\}$ ;

$\Delta t$  – продолжительность НТ,  $\Delta t \in \Delta T$ ;

$\mu$  – функция принадлежности участка НВР, ограниченного интервалом  $\Delta t$ , НТ  $\tau$ .

Классифицируем НТ НВР в зависимости от продолжительности на элементарную  $T \in \mathfrak{T} (\Delta t = 1)$ , локальную

$NT \in N\mathfrak{T} (1 < \Delta t < n - 1)$  и основную (общую)  $GT \in G\mathfrak{T} (\Delta t = n - 1)$ .

ВР НЭТ представим в виде (здесь  $TTend, RTend$  – процедуры получения нечетких множеств  $\tilde{v}_i$  и  $\tilde{a}_i$ ):

$$\begin{aligned} \tilde{v}_i &= TTend(\tilde{x}_i, \tilde{x}_{i+1}), \\ \tilde{a}_i &= RTend(\tilde{x}_i, \tilde{x}_{i+1}), \\ \mu_i &= \min(\mu(\tilde{x}_i), \mu(\tilde{x}_{i+1})). \end{aligned} \quad (3)$$

Уровни ВР описаны компонентами НТ – тип и интенсивность:  $\tau = \{t_i, \tau_i\}, i \in [1, n]$ , где  $\tau = \langle \tilde{v}, \tilde{a}, \Delta t, \mu \rangle$ .

Для выделения доминирующей тенденции определяем кластеры  $S = \{s_1, s_2, \dots, s_k\}$  подобных нечетких меток  $\tilde{x}_i$  и для каждого кластера  $s \in S$  определяем мощность.

Для нечеткой кластеризации тенденций используем FCM-алгоритм, основанный на минимизации целевой функции:

$$J = \sum_{j=1}^C \sum_{i=1}^N \mu_{ij}^m \|x_i - c_j\|, \quad (4)$$

где  $N$  – количество показателей;

$C$  – количество кластеров;

$\mu_{ij}$  – степень принадлежности объекта  $i$  кластеру  $j$ ;

$m$  – любое действительное число, большее 1;

$x_i$  –  $i$ -я тенденция множества тенденций ВР;

$c_j$  –  $j$ -й кластер;

$\|x_i - c_j\|$  – норма, характеризующая расстояние от центра кластера  $j$  до тенденции  $i$ .

Для выявления степени подобия предиката ВР2 НЭТ (введение новых функций) с заданным ВР1 НЭТ (количество ошибок) предлагается установить следующий критерий оценки соответствия поведения исследуемого ВР1 НТ  $\tau_1 = \langle \tilde{v}_1, \tilde{a}_1, \Delta t, \mu_1 \rangle$  поведению ВР2 предиката НТ  $\tau_2 = \langle \tilde{v}_2, \tilde{a}_2, \Delta t, \mu_2 \rangle$ , который определяет расстояние  $\rho(\tau_1, \tau_2)$  между одновременными НЭТ [9]:

$$\rho(\tau_1, \tau_2) = \frac{1}{n} \sum_{i=1}^n \left( \rho(\tilde{v}_i^{\tau_1}, \tilde{v}_i^{\tau_2}) \cdot \rho(\tilde{a}_i^{\tau_1}, \tilde{a}_i^{\tau_2}) \cdot \sqrt{\rho(\mu_i^{\tau_1}, \mu_i^{\tau_2})^2} \right),$$

где

$$\rho(\tilde{v}_i^{\tau_1}, \tilde{v}_i^{\tau_2}) = \begin{cases} 0, & \tilde{v}_i^{\tau_1} = \tilde{v}_i^{\tau_2}; \\ 1, & \tilde{v}_i^{\tau_1} \neq \tilde{v}_i^{\tau_2}, \tilde{v}_i^{\tau_1} \neq C; \\ 0.5, & \tilde{v}_i^{\tau_1} \neq \tilde{v}_i^{\tau_2}, \tilde{v}_i^{\tau_1} = C, \end{cases}$$

$$\rho(\tilde{a}_i^{\tau_1}, \tilde{a}_i^{\tau_2}) = \begin{cases} 1, & \tilde{a}_i^{\tau_1} \neq \tilde{a}_i^{\tau_2}; \\ 0, & \tilde{a}_i^{\tau_1} = \tilde{a}_i^{\tau_2}, \end{cases}$$

$$\rho(\mu_i^{\tau_1}, \mu_i^{\tau_2}) = |\mu_i^{\tau_1} - \mu_i^{\tau_2}|,$$

$$0 \leq \rho(\tau_1, \tau_2) \leq 1,$$

$\rho(\tilde{v}_i^{\tau_1}, \tilde{v}_i^{\tau_2})$  – функция расстояния типов НЭТ;

$\rho(\tilde{a}_i^{\tau_1}, \tilde{a}_i^{\tau_2})$  – функция расстояния интенсивностей

НЭТ;

$\rho(\mu_i^{\tau_1}, \mu_i^{\tau_2})$  – функция расстояния принадлежностей НЭТ.

Тенденция типа  $\tilde{v} = C$  противоположной тенденции не имеет.

Рассмотрим алгоритм прогнозирования ВР  $Y = \{t, x_t\}$ , ( $i = 1, 2, \dots, n$ ) при условии существования правдоподобности экспертной гипотезы, что НТ ВР  $Z = \{t, z_t\}$ , ( $i = 1, 2, \dots, k$ ) является предикатом ВР  $Y$  [10–11]. Алгоритм включает три шага:

1. Прогнозируются НЭТ ВР  $Y$ :  $\tau_{t+1}^Y = f(\tau_t^Y)$ , где  $\tau_{t+1}^Y$  – прогнозная НЭТ ВР  $Y$ ,  $\tau_t^Y$  – текущая НЭТ ВР  $Y$ ,  $f$  – зависимость в НЭТ ВР  $Y$ .

2. Прогнозная НЭТ ВР  $Y$  регулируется с учетом компонент основных тенденций исследуемого ВР  $G\tau_Y$  и ВР-предиката  $G\tau_Z$  соответственно:  $\hat{\tau}_{t+1}^Y = r(\tau_{t+1}^Y, G\tau_Y, G\tau_Z)$ , где  $\tau_{t+1}^Y$  – прогнозная НЭТ ВР  $Y$ ,  $\hat{\tau}_{t+1}^Y$  – прогнозная НЭТ ВР  $Y$  после корректировки,  $G\tau_Y$  – основная НТ ВР  $Y$ ,  $G\tau_Z$  – основная НТ ВР  $Z$ ,  $r$  – правила корректировки.

3. Получение оценки прогнозного значения числового ВР  $Y$ .

На этой основе предлагается следующий алгоритм прогнозирования краткосрочных НТ ВР.

Шаг 1. Преобразование числового ВР в НВР.

Шаг 2. Преобразование НВР в НВР НЭТ:

$$\tau_t^Y = \langle \tilde{v}_t, \tilde{a}_t, \mu_t \rangle.$$

Шаг 3. Построение модели изменения компонент НЭТ ВР  $Y$  и их прогнозирование на один период:

$$\tilde{v}_{t+1} = \tilde{v}_t \times \tilde{v}_{t-1} \times \dots \times \tilde{v}_{t-p} \circ R_v(t, t-p),$$

$$\tilde{a}_{t+1} = \tilde{a}_t \times \tilde{a}_{t-1} \times \dots \times \tilde{a}_{t-q} \circ R_a(t, t-q).$$

Шаг 4. Прогнозирование числового ВР  $Y$  с предварительной дефаззификацией компонент НТ

$$\tau_{t+1}^Y = \langle \tilde{v}_{t+1}, \tilde{a}_{t+1}, \mu_{t+1} \rangle, x_{t+1} = x_t + v_{t+1} \cdot a_{t+1}.$$

Шаг 5. Применение алгоритма идентификации основной тенденции (см. пункт 3, Алгоритм 1) для ВР  $Y$  и вычисление ее компонент  $G\tau_Y = \langle \tilde{v}_{G\tau}^Y, \tilde{a}_{G\tau}^Y, \mu_{G\tau}^Y \rangle$ . Дефаззификация компонент основной НТ ВР  $Y$ .

Шаг 6. Шаг 5 для ВР  $Z$ .

Шаг 7. Регулировка прогнозной НЭТ ВР  $Y$ :

$$\hat{\tau}_{t+1}^Y = r(\tau_{t+1}^Y, G\tau_Y, G\tau_Z),$$

$$\hat{\tau}_{t+1}^Y = v_{t+1} \cdot a_{t+1} + v_{G\tau}^Y \cdot a_{G\tau}^Y + v_{G\tau}^Z \cdot a_{G\tau}^Z.$$

Шаг 8. Вычисление отрегулированного прогнозного значения числового ВР  $Y$  на один период:

$$x_{t+1} = x_t + \hat{\tau}_{t+1}^Y.$$

### 3 СРЕДСТВА АНАЛИЗА И УПРАВЛЕНИЯ ПРОЕКТАМИ

Всю информацию о процессе реализации проекта компания сохраняет в различных направлениях: разработчики; метки всевозможных состояний проекта, каждая из которых имеет шифр, дату, имя разработчика, код программы; выпущенные и запланированные релизы с номером версии, датой и функциональными характеристиками.

Для получения статистики и прогноза по текущему состоянию проекта предлагается использование компоненты анализа данных временных рядов работы над проектом, полученных посредством системы контроля версий (VCS).

Чтобы получить анализ выбранных временных рядов, выгружаем их из СКВ в файл формата .xls и загружаем их в программу экспресс-анализатора, в которой выделяется доминирующая тенденция с помощью кластеризации ряда, высчитываются подобие исследуемых рядов и наличие между ними корреляции [12]. Далее посредством трех гипотез получаем прогноз зависимого ряда, при наличии явного подобия данных используем полученный анализ ряда предиктора (который имеет влияние на исследуемый ряд) для корректировки прогноза. Учитывая ошибки прогноза, подсчитанные в экспресс-анализаторе, и знания эксперта (в лице руководителя проекта), выбираем прогноз наиболее подходящей гипотезы. На основе прогноза и экспертного заключения будет получен ориентир на стабильную версию последующего релиза, выявлена успешность проекта, что позволит руководителю верно распределять задания сотрудникам. Предлагаемая схема представлена на рисунке 1.

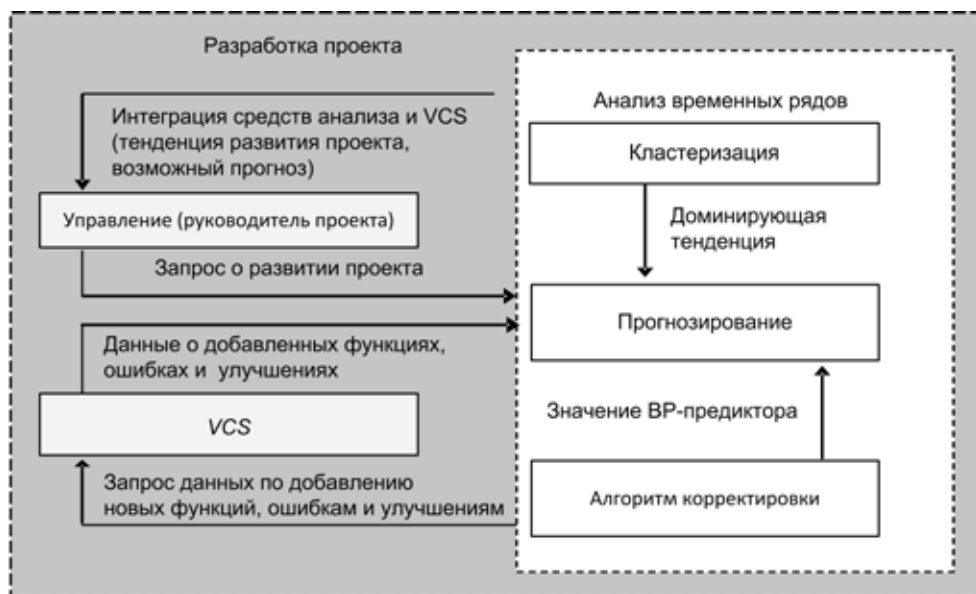


Рис. 1. Разработка проекта с введением средств анализа для управления

#### 4 ПРИМЕР РЕАЛИЗАЦИИ ПРЕДЛОЖЕННОГО ПОДХОДА

В качестве примера был рассмотрен проект MongoDB, а именно главная его программная часть Core Server. Данные об ошибках, улучшениях и нововведенных функциях были взяты с ресурса [www.jira.mongodb.org](http://www.jira.mongodb.org).

MongoDB – документно-ориентированная система управления базами данных (СУБД) с открытым исходным кодом, не требующая описания схемы таблиц. Написана на языке C++.

ВР были взяты за период начиная с декабря 2014 по август 2015.

В таблице 1 представлены результаты анализа.

По полученным результатам анализа можно сделать следующие выводы: по статистике работы над проектом добавленные улучшения и введенные новые функции имеют значительные сходство и корреляцию, это говорит о влиянии результатов разработки новых функциональных возможностей на улучшения в проекте. Аналогичны выводы по появившимся ошибкам и улучшениям.

Рассмотрим подробнее результаты прогнозирования введенных улучшений на основе гипотезы сохранения тенденции с учетом влияния добавленных новых функций.

График Improvement – New Feature работы над проектом с учетом гипотезы сохранения тенденции представлен на рисунке 2.

В таблице 3 представлен прогноз появления Improvement.

На рисунке 2 представлен вариант прогнозирования с использованием гипотезы, где ось  $x$  – значение ВР, ось  $t$  – время; « $\blacklozenge$ » обозначает ВР  $Y$  (Improvement); « $\blacktriangle$ »

Анализ данных проекта

Тип метки	Общая тенденция	Доминирующая тенденция	Мера сходства	Корреляция	Интерпретация корреляции
Bug	Падение	Стабильность	0,52	-0,2026	Слабая
Improvement	Рост	Стабильность			
Improvement	Рост	Стабильность	0,61	0,5581	Средняя
New Feature	Рост	Рост Сильный			
Bug	Падение	Стабильность	0,52	0,1778	Слабая
New Feature	Рост	Рост Сильный			

Таблица 2

Результаты прогнозирования с учетом влияния ВР-предиката

ВР зависимый	ВР-предикат	Гипотеза 1	Гипотеза 2	Гипотеза 3
Improvement	Bug	Стабильность	Рост сильный	Стабильность
Improvement	New Feature	Рост средний	Рост сильный	Рост сильный
Bug	Improvement	Стабильность	Падение сильное	Стабильность
Bug	New Feature	Рост средний	Падение сильное	Рост сильный

Таблица 3

Значения Improvement по результатам прогноза

Гипотеза	Значение Improvement			Тенденция, интенсивность ( $\hat{\tau}_{t+1}^Y$ )
	$\tau_{t+1}$	$\tau_{t+1}^Y$	$\hat{\tau}_{t+1}^Y$	
Гипотеза сохранения тенденции	0,1143	0,1617	0,5744	Рост средний
Гипотеза устойчивости тенденции	0,7857	0,8331	1,2459	Рост сильный
Гипотеза прогноза на заданный период	0,2857	0,3331	0,7459	Рост сильный

Таблица 4

Оценка качества прогноза (Improvement - New Feature)

Гипотеза	MSE			RMSE		
	$\tau_{t+1}$	$\tau_{t+1}^Y$	$\hat{\tau}_{t+1}^Y$	$\tau_{t+1}$	$\tau_{t+1}^Y$	$\hat{\tau}_{t+1}^Y$
Гипотеза сохранения тенденции	0,209	0,1679	0,0052	0,4571	0,4097	0,072
Гипотеза устойчивости тенденции	0,209	0,2546	0,9728	0,4571	0,5046	0,0992
Гипотеза прогноза на заданный период	0,0018	0,0278	0,0992	0,0429	0,1669	0,3149

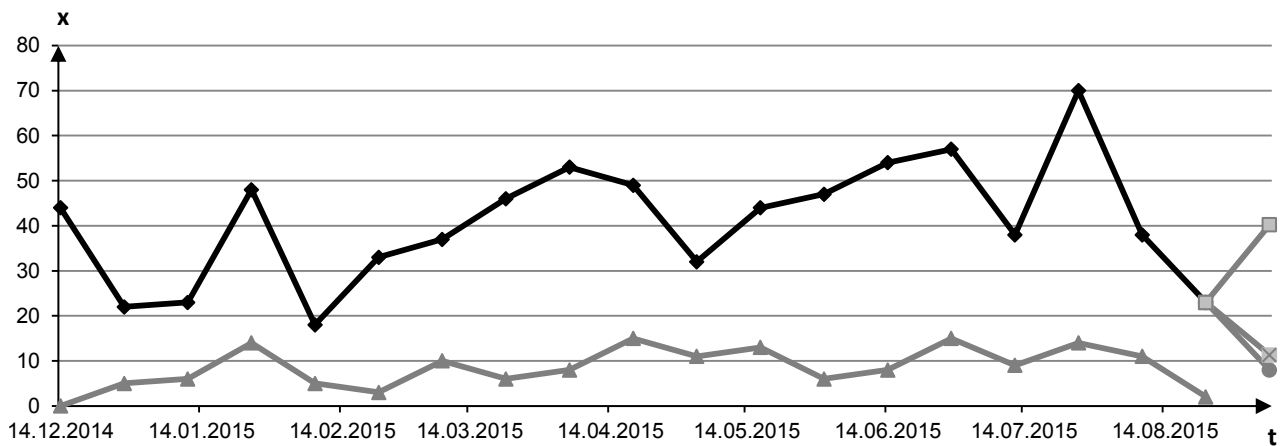


Рис. 2. Прогнозирование появления улучшений в проекте MongoDB с учетом влияния добавления новых функциональных возможностей в проект (гипотеза сохранения тенденции)

обозначает BP Z (New Feature); «●—» обозначает прогноз без учета корректировки; «×—» обозначает прогноз после корректировки BP Y по основной тенденции BP Y; «■—» обозначает прогноз после корректировки BP Y по основной тенденции BP Z.

Для получения надежности прогноза использованы квадрат среднеквадратичной ошибки (Mean Square Error, MSE) и среднее квадратичная ошибка (Root Mean Square Error, RMSE). Результаты оценки качества прогнозирования улучшений в разработке проекта с учетом влияния добавленных новых функций в проект представлены в таблице 4.

Согласно предложенному алгоритму прогнозирования на первой фазе при прогнозировании по гипотезе сохранения тенденции BP1 Y (Improvement) (рис. 2) получили прогнозную НЭТ «Рост средний» и прогнозные значения BP (MSE=0,209). Затем на второй фазе, применяя алгоритм 1, определили основную доминирующую НТ BP1 Y «Рост средний» (табл. 2) и основную НТ BP2 Z (New Feature) «Рост сильный» при общей тенденции «Рост», дефазифицированные значения которых были использованы для коррекции прогнозных значений BP1 Y, полученного на первой фазе (MSE= 0,0052) (см. табл. 4).

### ЗАКЛЮЧЕНИЕ

В существующих системах управления версиями проектов находится множество различных метрик ведения работы над проектом: commit, release, bug, improvement, new feature, line, branch, epic, question, spec, story, task, sub-task и т.д., однако они не ориентированы на анализ и прогнозирование с учетом того, что показатели работы над проектом могут содержать нелинейные, разнонаправленные и неоднородные данные и связи. Поэтому для реализации прогнозирования метрик программного кода системы контроля версий был выполнен обширный анализ показателей проектных работ.

Полученные результаты данного примера показывают, что предложенный подход, реализующий модифицированный метод прогнозирования НЭТ, может быть использован для краткосрочного прогнозирования BP в ситуации, когда имеется экспертное предположение о существовании BP-предиката.

### СПИСОК ЛИТЕРАТУРЫ

1. Сайт компании Википедия. – Режим доступа: <https://ru.wikipedia.org/> (дата обращения: 10.08.2015).
2. Сайт компании GitHub. – Режим доступа: <https://github.com/> (дата обращения: 10.08.2015).
3. Сайт компании Git-SCM. – Режим доступа: <https://git-scm.com/book/ru/v1> (дата обращения: 15.08.2015).
4. Бокс Дж., Дженкинс Г. Анализ временных рядов. Прогноз и управление: пер. с англ. / под ред. В. Ф. Писаренко. – М.: Мир, 1974. – 406 с.
5. Кендэл М. Временные ряды/ пер. с англ. и пред. Ю. П. Лукашина. – М.: Финансы и статистика, 1981. – 199 с.
6. Lotfi A. Zadeh. Fuzzy Sets. Information and Control. – 1965.
7. Song Q. and B. Chissom. Fuzzy time series and its models. Fuzzy Sets and Systems. – 1993, no. 54, – pp. 269–277.
8. Ярушкина Н.Г., Афанасьева Т.В., Перфильева И.Г. Интеллектуальный анализ временных рядов: учебное пособие. – Ульяновск: УлГТУ, 2010. – 320 с.
9. Тимина И.А. Нечеткая зависимость как метод решения задач интеллектуального анализа временных рядов // Автоматизация процессов управления. – 2013. – № 3(33). – С. 39–44.
10. Yarushkina N., Afanasieva T., Timina I. Predicative analytics for developing software // SYRCoSE 2014, The 8th Spring / Summer Young Researchers' Colloquium on Software Engineering, May 29–31, St. Peterburg, Russia, pp. 154–158.

11. Ярушкіна Н.Г., Афанасьева Т.В., Романов А.А., Тимина И.А. Извлечение знаний о зависимостях временных рядов для задач прогнозирования // Радиотехника. – 2014. – № 7. – С. 141–146.

12. Свидетельство о государственной регистрации программы для ЭВМ №2014612213 от 21.02.2014. Интеллектуальный экспресс-анализатор сходства нечетких тенденций временных рядов.

## REFERENCES

1. *Sait kompanii Vikipediia* [Wikipedia Web Site]. Available at: <https://ru.wikipedia.org/> (accessed 10.08.2015).

2. *Sait kompanii GitHub* [GitHub Web Site]. Available at: <https://github.com/> (accessed 10.08.2015).

3. *Sait kompanii Git-SCM* [Git-SCM Web Site]. Available at: <https://git-scm.com/book/ru/v1> (accessed 15.08.2015).

4. Box G., G. Jenkins. *Analiz vremennykh riadov. Prognoz i upravlenie*. Per. s angl. pod red. V.F. Pisarenko [Time Series Analysis: Forecasting and Control. Translated from Engl. ed. by V.F. Pisarenko]. Moscow, Mir Publ., 1974. 406 p.

5. Candle M. *Vremennye riady*. per. s angl. i pred. Yu.P. Lukashina [Time Series Translated from Engl. Yu.P. Lukashin]. Moscow, Finansy i statistika Publ., 1981. 199 p.

6. Lotfi A. Zadeh. *Fuzzy Sets. Information and Control*. 1965.

7. Song Q., Chissom B. Fuzzy Time Series and its Models. *Fuzzy Sets and Systems*, no. 54, 1993, pp. 269–277.

8. Yarushkina N.G., Afanasieva T.V., Perfilieva I.G. *Intellektualnyi analiz vremennykh riadov: uchebnoe posobie* [Intellectual Analysis of Time Series. Textbook]. Ulyanovsk, ULSTU Publ., 2010. 320 p.

9. Timina I.A. Nechetkaia zavisimost kak metod resheniia zadach intellektualnogo analiza vremennykh riadov [Fuzzy Dependency as a Problem-Solving Method for Time-Series Analysis]. *Avtomatizatsiia protsessov upravleniia* [Automation of Control Processes], 2013, no. 3 (33), pp. 39–44.

10. Yarushkina N., Afanasieva T., Timina I. Predicative analytics for developing software. *SYRCoSE 2014, the 8th Spring. Summer Young Researchers' Colloquium on Software Engineering*, St. Peterburg, Russia, May, 29–31, 2014, pp. 154–158.

11. Yarushkina N.G., Afanasieva T.V., Romanov A.A., Timina I.A. Izvlechenie znaniia o zavisimostiakh vremennykh riadov dlia zadach prognozirovaniia [The Extraction of Knowledge about Dependencies of Time Series for the Tasks of Forecasting]. *Radiotekhnika* [Radioengineering], 2014, no. 7, pp. 141–146.

12. *Intellektualnyi ekspress-analizator skhodstva nechetkikh tendentsii vremennykh riadov*. [Intelligent Express-Analyzer of the Similarity of Time Series Fuzzy Tendency]. Certificate of State Registration RF for Computer Program, no. 2014612213, dated 21.02.2014.