

УДК 359.07  
DOI: 10.15827/0236-235X.120.706-710

Дата подачи статьи: 13.05.17  
2017. Т. 30. № 4. С. 706–710

## **РАЦИОНАЛЬНАЯ ОРГАНИЗАЦИЯ ПРОЦЕССА РАЗРАБОТКИ ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ КАК ПРЕДПОСЫЛКА УСПЕШНОЙ АВТОМАТИЗАЦИИ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ**

*О.В. Тиханьчев, к.т.н., старший научный сотрудник, tow65@yandex.ru;*

*А.В. Макарецев, начальник отдела, tvladlen@inbox.ru;*

*В.Р. Гахов, зам. начальника отдела, начальник лаборатории, gahovvlad@yandex.ru*

*(27 Центральный научно-исследовательский институт Минобороны России,  
1-й Хорошевский проезд, 5, г. Москва, 123007, Россия)*

Несмотря на достаточно длительный период практического применения программных продуктов и развития средств разработки ПО, сам процесс их разработки до настоящего времени оптимизирован недостаточно. Прежде всего это касается разработки прикладного ПО.

Как показывает анализ предметной области, существенная часть проблем создания качественного ПО обусловлена организацией разработки прикладных программ.

В настоящее время существует достаточно широкий спектр методологий разработки ПО и инструментов для их программной реализации: Rational Unified Process, Microsoft Solution Framework, Extreme Programming, Agile, Capability Maturity Model Integration.

В статье рассмотрены возможные подходы к рациональной организации данного процесса с использованием специализированных программных систем, таких как Team Foundation Server, на основе стандарта управления проектами Project Management Body of Knowledge (PMBOK). Сделаны выводы об объективной необходимости использования для решения организационно-технических проблем создания ПО средств автоматизированной организации разработки. В то же время использование только системы TFS, как и любого другого продукта организации управления проектами, не решает всех организационных проблем.

Для успешной реализации применения подобных систем необходимо организовать взаимодействие всех участников процесса, обеспечить подготовку и внедрение в процесс разработки программ всех категорий специалистов. Как показывает мировая практика, реализация предлагаемых мер может существенно повысить эффективность разработки и сопровождения жизненного цикла ПО, обеспечить переход от существующих ремесленных подходов к программированию к промышленной разработке программ.

**Ключевые слова:** *прикладное ПО, разработка ПО, автоматизация поддержки принятия решений, программная инженерия, Team Foundation Server.*

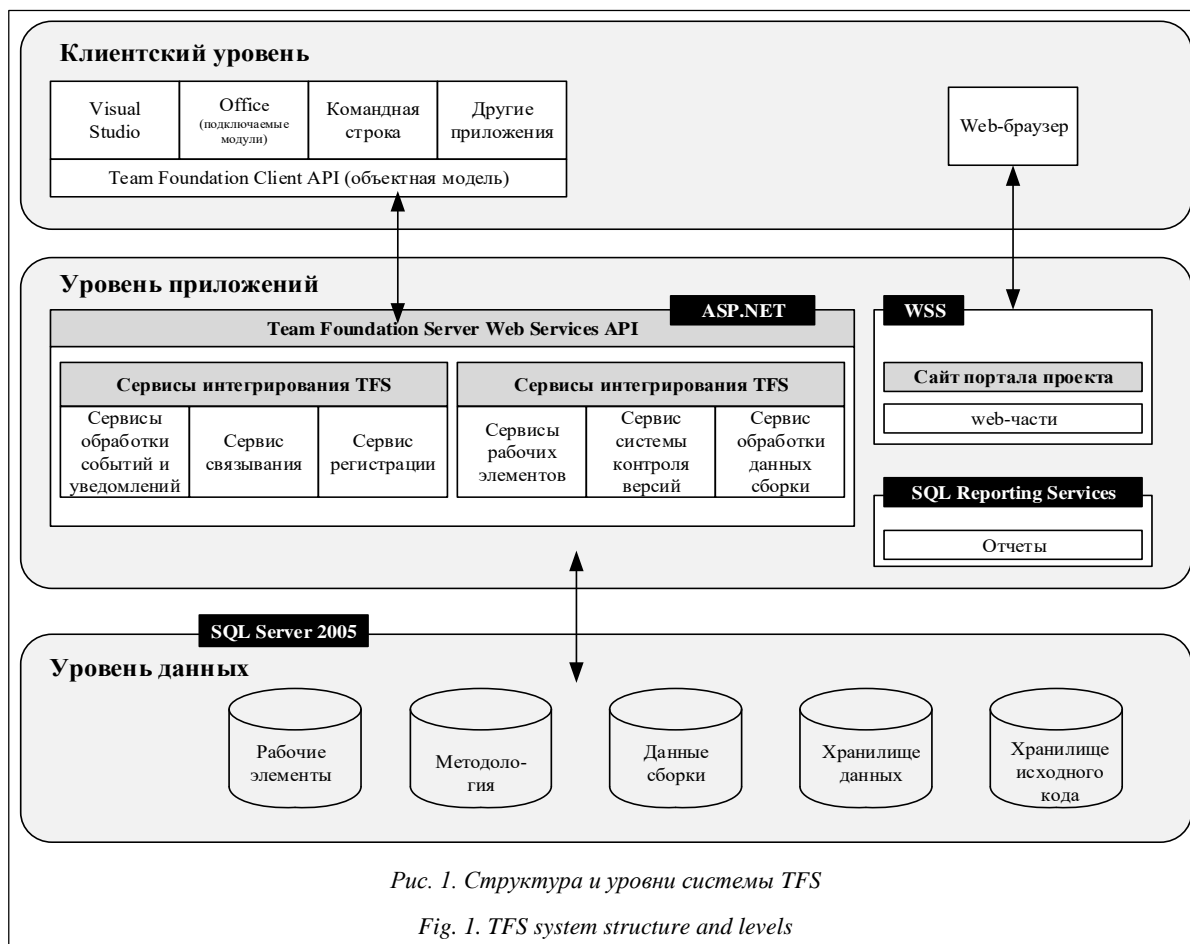
Один из важнейших компонентов любой автоматизированной системы поддержки принятия решений – ее ПО. В подавляющем большинстве случаев именно от него зависит качество вырабатываемых управленческих решений [1]. Но до настоящего времени в процессе создания прикладного ПО имеются определенные проблемы: недостаточный уровень совместимости разрабатываемых программ, дублирование работ, большие затраты времени на тестирование [2]. Большинство из этих проблем организационные. На первый взгляд, логично искать их решение через реализацию организационных мер, как и делается в настоящее время. Однако преодоление проблем исключительно путем задания соответствующих требований в документах, определяющих разработку ПО, не всегда эффективно, особенно при разработке структурно-сложных систем большими территориально распределенными коллективами разработчиков. Впрочем, как показывает опыт, проблема может быть решена с помощью соответствующего инструментария.

### **Использование специализированных программных средств управления проектами**

Как показывает мировая практика разработки ПО, эффективные пути решения организационных

проблем существуют не только в части качественной разработки задающих документов. Альтернативу можно найти в области организации труда всех категорий разработчиков за счет использования как структурных подходов на основе стандартов управления проектами PMBoK (Project Management Body of Knowledge), ISO 21500 и реализующей их положения методологии организации разработки RUP (Rational Unified Process), так и специализированного инструментария автоматизации управления проектами – Jira, Gemini, Savana, Redmine, Trac, TaskJuggler, TFS и им подобных. Перечисленные системы близки по своим функциональным возможностям, каждая имеет преимущества и недостатки, определяющие приоритетность их выбора теми или иными группами пользователей. В данном случае не столь важно, как называется продукт или кто его разработчик, главное – функционал, а именно – возможность автоматизированной рационализации процесса разработки программной продукции.

Возможности подобного инструментария целесообразно раскрыть на примере одной из таких систем – TFS (Team Foundation Server), предназначенной именно для совместной распределенной работы над проектами по созданию компонентов ПО. Система TFS, являющаяся дальнейшим развитием программы VSS (Microsoft Visual SourceSafe),



представляет собой комплексное решение по управлению проектами разработки ПО.

Построена TFS по принципу трехуровневой архитектуры (рис. 1).

Клиентский уровень системы используется для создания, администрирования и управления проектами, а также для доступа разработчиков к элементам проекта. Указанные операции осуществляются через следующие компоненты TFS:

- объектная модель TFS, представляющая собой открытый API, обеспечивающий взаимодействие TFS и внешних программ;
- компоненты VSIP (Visual Studio Industry Partners), инструментальные средства, надстройки и компоненты заимствованных языков программирования для Visual Studio;
- компоненты интеграции с офисными системами, обеспечивающие возможность запроса и обновления рабочих элементов в БД TFS Work Item Tracking;
- инфраструктурные компоненты обеспечения политики регистрации изменений файлов в системе контроля версий.

Уровень приложений, или прикладной уровень, представляет собой web-сервисы ASP.NET, взаимодействующие с клиентским уровнем системы. Компоненты прикладного уровня сгруппированы по сервисам:

- обработки данных Team Foundation Data Services;
- интеграции Team Foundation Integration Services;
- организации информационного обмена Windows Sharepoint Services (WSS);
- формирования отчетов SQL Reporting Services.

В состав прикладных сервисов входят средства контроля версий TFVC (Team Foundation Version Control), серверы сборки Team Build и отладки Team Load Test Agents, портал командного проекта Team Project Portal и другие.

Уровень данных базируется на SQL Server 2005 Standard Edition и обеспечивает сервисы хранения данных и документов. Он состоит из программных компонентов и хранилищ данных:

- компонентов статуса рабочих элементов;
- единого репозитория на базе SharePoint, содержащего документацию на разрабатываемое ПО.

### Практика применения системы TFS для управления разработкой программных средств

Работа с TFS организуется в соответствии с логикой типового процесса разработки ПО: создаются проекты (Team Projects), в составе проектов

задаются последовательно или параллельно выполняемые работы, состоящие из связанных наборов задач для исполнителей. Работы и задачи обладают определенным набором свойств и статусов (рис. 2). Свойства позволяют описать содержание задачи, назначить исполнителя, указать сроки реализации и т.п. Статусы, связи между задачами и свойствами позволяют контролировать время и полноту выполнения этапов проекта.

Применение TFS обеспечивает автоматизированное выполнение целого ряда частных функций управления проектами:

- многократную блокировку файлов для изменения, обеспечивающую параллельную работу нескольких пользователей;
- отложенное внесение изменений;
- ветвление процесса с последующим слиянием и урегулированием возможных конфликтов в случае слияния различных веток проекта;
- разграничение уровней доступа с детализацией до файла (каталога);
- поддержку версии документации;
- обеспечение откатов до предыдущих версий, в том числе по отдельным веткам проекта;
- реализацию операций подтверждения малых изменений.

Как показал опыт использования TFS при разработке прикладного ПО, данное средство через реализацию перечисленных ранее частных функций обеспечивает автоматизацию основных составляющих процесса разработки:

- постановку задач группам и отдельным разработчикам, контроль их выполнения;
- архивацию данных с контролем версии и вносимых в код изменений;
- сборку версий, в том числе автоматическую, с рассылкой уведомлений;

– отладку разрабатываемых программ, в том числе с настройками автоматического режима процесса тестирования.

Не преувеличивая возможности TFS и не занижая качества других подобных систем, можно сделать вывод, что с учетом их функционала автоматизация процесса организации разработки на основе использования специализированного инструментария обеспечивает существенное повышение его эффективности.

### Возможные преимущества автоматизации управления проектами

Использование TFS, как и других подобных средств, не решает всех проблем в области разработки ПО. Можно отметить, что при грамотной организации руководства и наличии отлаженного процесса организации работы все функции управления процессом разработки могут быть реализованы и в неавтоматизированном режиме [3]. Это касается и работы над небольшими уникальными проектами, разрабатываемыми в сжатые сроки для конкретной функции управления. Хотя последний случай все-таки спорный. Быстро разработанный и внедренный проект приходится дорабатывать в ходе эксплуатации, и использование TFS сильно упростило бы процесс настройки под требования пользователей и раннее выявление ошибочных решений. Но для некоторой среднестатистической организации, параллельно ведущей несколько проектов, TFS – серьезное подспорье в организации разработки. Несомненным преимуществом использования средств управления проектами станет, с одной стороны, контроль всех этапов и направленной деятельности по проекту, с другой, из-за оптимизации управления в целом сократится время разработки и, соответственно, его стоимость, а также

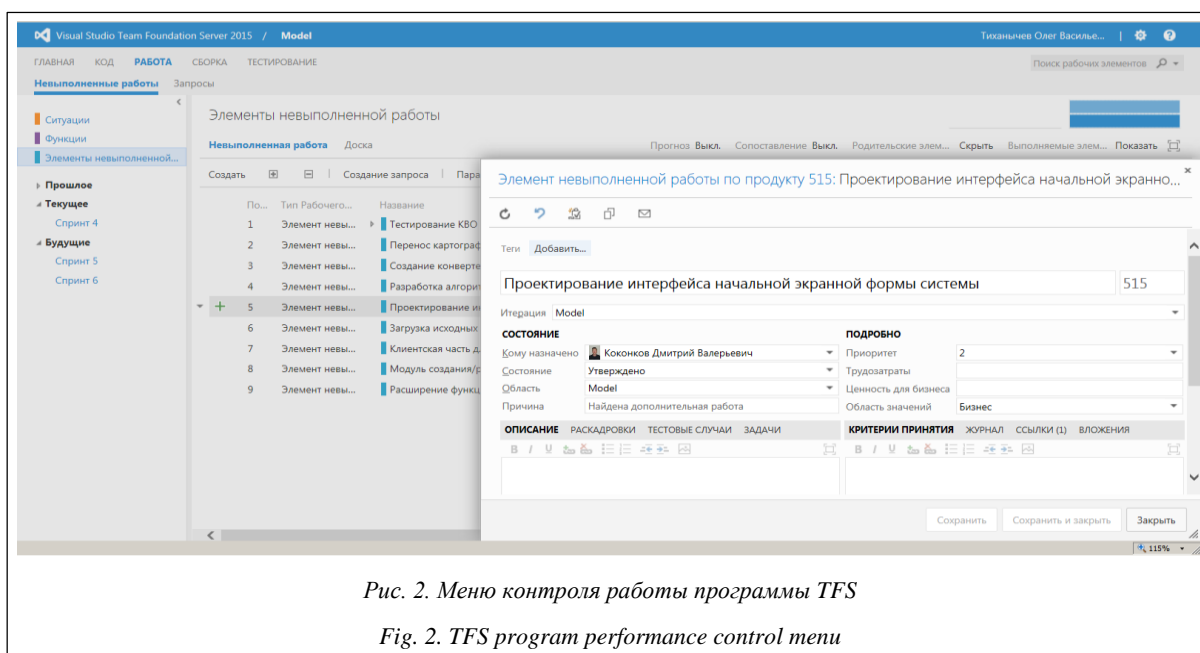


Рис. 2. Меню контроля работы программы TFS

Fig. 2. TFS program performance control menu

повысится качество конечного изделия. Дополнительный бонус – данный тип ПО является технологическим, его влияние на состояние конечного программного продукта минимально, и, исходя из этого, его применение не противоречит требованиям информационной безопасности.

### Заключение

Как уже отмечалось ранее, TFS – далеко не единственная система организации разработки программных продуктов. Существует достаточно большое количество аналогичных программных средств, имеющих те или иные преимущества и недостатки. Использование конкретной системы – выбор разработчика, определяемый его предпочтениями, требованиями и условиями ведения проекта.

Разумеется, использование средств автоматизированной организации процесса разработки ПО – не панацея. Всех проблем ни RUP, ни TFS, ни любое другое средство или стандарт сами по себе не решат. Это касается и других методологий, ведь, кроме RUP, которую специалисты считают одной из наиболее рациональных, могут использоваться стратегии Microsoft Solution Framework (MSF), Extreme Programming (XP), Agile, Capability Maturity Model Integration (CMMI) и другие. Каждая из этих методологий имеет собственные подходы к построению процессов разработки и средства их организации. Целесообразность применения любой из них зависит от масштабов и характера проекта, порядка его финансирования, апробирования инструментария конкретным коллективом. Важна не конкретная методика, а сам факт упорядочения процесса разработки для достижения эффективного конечного результата. И применение любой из них – одна из предпосылок повышения эффективности использования интеллектуального потенциала страны и перехода от использования зарубежного ПО к отечественному, что весьма актуально в условиях перманентного информационного противоборства в современном постиндустриальном мире [4, 5]. В то же время практическое

внедрение систем автоматизированной организации разработки ПО – это только первый шаг по совершенствованию данного процесса.

Для практической реализации этого и последующих шагов требуется целый комплекс мер. Например, наличие необходимых категорий разработчиков: программистов, программных инженеров, алгоритмистов, аналитиков, руководителей проектов, подготовленных с учетом современных требований [6, 7], и ряд других мер.

Результатом реализации предлагаемых мер станет комплексный подход к разработке, который создаст предпосылки для перехода от ремесленных методов разработки программ к современным технологиям промышленного программирования [8, 9].

### Литература

1. Выпасняк В.И., Тиханычев О.В. Автоматизированные системы управления войсками (силами): тенденции, методы и перспективы развития // Вестн. Акад. воен. наук. 2009. № 4. С. 61–68.
2. Тиханычев О.В., Саяпин О.В., Гахов В.Р. Использование современных технологий информационного обследования как фактор эффективности автоматизации управления // Информатизация и связь. 2013. № 6. С. 90–93.
3. Иванова. Г.С., Ничушкина Т.Н. Проектирование программного обеспечения. М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. 74 с.
4. Выпасняк В.И., Тиханычев О.В., Гахов В.Р. Киберугрозы автоматизированным системам управления // Вестн. Акад. воен. наук. 2013. № 1. С. 103–109.
5. Рыбина Г.В. Проблемы интеграции и особенности разработки программного обеспечения современных интеллектуальных систем // Прикладная физика и математика. 2013. № 3. С. 41–63.
6. Летбридж Т. [и др.]. SE2004: рекомендации по обучению специальности «Программная инженерия» // Открытые системы. СУБД. 2006. № 10. URL: <http://www.osp.ru/os/2006/10/3910113> (дата обращения: 10.04.2017).
7. Рекомендации по преподаванию программной инженерии и информатики в университетах; [пер. с англ.]. М.: ИНТУИТ.РУ, 2007. 462 с.
8. Архипов Л.В. Концепция применения контактно-адресуемых систем для хранения и контроля версий // Инновации в науке. 2015. № 52-1. С. 42–46.
9. Архипов Л.В. Концепция ветвления процесса разработки проекта в системах контроля версий // Науч. альманах. 2015. № 12-2. С. 22–25.

### RATIONAL ORGANIZATION OF AN APPLICATION SOFTWARE DEVELOPMENT PROCESS FOR DECISION SUPPORT SUCCESSFUL AUTOMATION

**O.V. Tikhanychev**<sup>1</sup>, Ph.D. (Engineering), Senior Researcher, [tow65@yandex.ru](mailto:tow65@yandex.ru)

**L.V. Makartsev**<sup>1</sup>, Head of Department, [mvladlen@inbox.ru](mailto:mvladlen@inbox.ru)

**V.R. Gakhov**<sup>1</sup>, Deputy Head of Department, Head of Laboratory, [gahovvlad@yandex.ru](mailto:gahovvlad@yandex.ru)

<sup>1</sup> 27 Central Research Institute of the Ministry of Defense of Russia, 1st Khoroshevsky lane 5, Moscow, 123007, Russian Federation

**Abstract.** Despite a long period of using software solution and development of software development tools, this process itself is still not optimized enough. This is especially important for application software development.

The subject area analysis shows that a significant part of the problems related to creating high-quality software is in the organization of application programs development. Currently, there is a wide range of methodologies for software development and software implementation tools: Rational Unified Process, Microsoft Solution Framework, Extreme Programming, Agile, Capability Maturity Model Integration.

The article describes possible approaches to rational organization of the above process using specific software systems, such as Team Foundation Server based on Project Management Body of Knowledge (PMBok). There are conclusions about the need for automated development organization solutions for organizational and technical issues of software development. At the same time, the only use of the TFS system, as any other product of project management organization, does not solve all organizational problems.

The paper states that for successful implementation of such systems it is necessary to organize interaction between all participants, to train and implement all specialists into a software development process. The world practice analysis shows that implementation of the proposed measures can significantly increase the effectiveness of software life cycle development and maintenance, providing a transition from existing “artisanal” programming approaches to industrial software development.

**Keywords:** application software, software development, decision support automation, software engineering, Team Foundation Server.

### References

1. Vypasnyak V.I., Tikhanychev O.V. The automated control of troops (forces): trends, methods and prospects of development. *Vestnik Akademii voennykh nauk* [Bulletin of the Academy of Military Sciences]. 2009, no. 4 (29), pp. 61–68 (in Russ.).
2. Tikhanychev O.V., Sayapin O.V., Gakhov V.R. The use of modern information technologies as a factor of the survey management automation efficiency. *Informatizatsiya i svyaz* [Informatization and Communications]. 2013, no. 6, pp. 90–93 (in Russ.).
3. Ivanova G.S., Nichushkina T.N. *Proektirovanie programmnoho obespecheniya* [Software Engineering]. Moscow, Bauman MSTU Publ., 2002, 74 p.
4. Vypasnyak V.I., Tikhanychev O.V., Gakhov V.R. Cyber threats automated control systems. *Vestnik Akademii voennykh nauk* [Bulletin of the Academy of Military Sciences]. 2013, no. 1 (42), pp. 103–109 (in Russ.).
5. Rybina G.V. Problems of integration and features of the software development of modern intellectual systems. *Prikladnaya fizika i matematika* [Applied Physics and Mathematics]. 2013, no. 3, pp. 41–63 (in Russ.).
6. Letbridzh T. SE2004: recommendations on educating the specialty “Software Engineering”. *Otkrytye sistemy. SUBD* [Open Systems. DBMS]. 2006, no. 10. Available at: <http://www.osp.ru/os/2006/10/3910113> (accessed April 10, 2017).
7. *Software Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering; Computing Curricula 2001: Computer Science*. Available at: <https://www.hse.ru/data/763/313/1234/SE2004Volume.pdf> (accessed July 29, 2017).
8. Arkhipov L.V. The concept of application contact-addressing systems for version storage and control. *Innovatsii v nauke* [Innovations in Science]. 2015, no. 52-1, pp. 42–46 (in Russ.).
9. Arkhipov L.V. The concept of project development process branching in version control systems. *Nauchny almanakh* [Scientific Almanac]. 2015, no. 12-2, pp. 22–25 (in Russ.).

### Примеры библиографического описания статьи

1. Тиханычев О.В., Макарецев Л.В., Гахов В.Р. Рациональная организация процесса разработки прикладного программного обеспечения как предпосылка успешной автоматизации поддержки принятия решений // Программные продукты и системы. 2017. Т. 30. № 4. С. 706–710. DOI: 10.15827/0236-235X.120.706-710.
2. Tikhanychev O.V., Makartsev L.V., Gakhov V.R. Rational organization of an application software development process for decision support successful automation. *Programmnye produkty i sistemy* [Software & Systems]. 2017, vol. 30, no. 4, pp. 706–710 (in Russ.). DOI: 10.15827/0236-235X.120.706-710.