

Тема 1. Создание облака ключевых слов англоязычного текста по профессиональной тематике

Подготовила: Пляскина Ульяна Сергеевна

1. Найдите текст на английском языке, посвященный разработке программных продуктов
https://www.developerdotstar.com/mag/articles/automate_software_process.html

2. Выберите фрагмент текста (не менее 15 предложений).

Automating repetitive procedures can provide real value to software development projects. In this article, we will explore the value of and barriers to automation and provide some guidance for automating aspects of the development process.

Although few experienced developers and project managers would argue the merits of automating development and testing procedures, when push comes to shove many teams place a low priority on implementing automated processes. The result is usually that, if automation is considered at all, it is given lip service early in the project life cycle, but falls quickly by the wayside.

Experience teaches us over and over again that trying to run a "simple" project by implementing a series of "simple" manual protocols, backed by "simple" written (sometimes, even just verbal) instructions, just doesn't work well. Even so, many of us still tend to allow ourselves to start the next project with the thought that the manual, protocol-based method will "do just fine."

After all, aren't we all professionals? Can't we read a set of simple instructions and just be disciplined enough to follow those instructions when the time is right? Isn't it a waste of time and money to invest in automating procedures for such a small project? The development team is only a half-dozen people after all—and the argument against automation goes on and on.

If you're a developer or tester who enjoys spending your time actually adding value to your project, rather than repeating the same routine tasks over and over, you'll want to consider advocating the concept of automation to your team (especially, to your project manager). If you're a project manager who's committed to maximizing the talents and time of the members of your technical team, as well as minimizing the risk of your project failing to deliver on time and on quality, you will want to encourage your team to invest the necessary time and effort required to automate the types of tasks that will be identified in this article.

Why Should I Automate?

You may already be familiar with many of the benefits of automating development processes. Some of the more commonly cited ones are:

Repeatability. Scripts can be repeated, and, unless your computer is having a particularly bad day, you can be reasonably certain that the same instructions will be executed in the same order each time the same script is run.

Reliability. Scripts reduce chances for human error.

Efficiency. Automated tasks will often be faster than the same task performed manually. (Some people might question whether gains in efficiency are typical, noting that they have worked on projects where, in their view, trying to automate tasks actually cost the project more time than it saved. Depending on the situation, this may be a real concern. In addition, automation might have been implemented poorly or carried too far on some projects—but keep reading for more on what to automate, and when.)

Testing. Scripted processes undergo testing throughout the development cycle, in much the same way the system code does. This greatly improves chances for successful process execution as the project progresses. Automated scripts eventually represent a mature, proven set of repeatable processes.

Versioning. Scripts are artifacts that can be placed under version control. With manual processes, the only artifacts that can be versioned and tracked are procedure documents. Versioning of human beings—the other factor in the manual process equation—is unfortunately not supported by your typical source control system.

Leverage. Another big benefit to automating is that developers and testers can focus on the areas where they add real value to a project—developing and testing new code and features—instead of worrying about the underlying development infrastructure issues.

For example, instead of requiring everyone to become intimately familiar with all the little nuances of the build procedure, you can have one person focus on automating the build and have that person provide the team with a greatly simplified method of building, hopefully as simple as running a command or two. Less time spent on builds leaves more time for the tasks that add the most value to the project.

3. Сделайте перевод этого фрагмента, используя две разные программы-переводчики (в качестве одной из них выберите Google Translate)

Google Translate

Автоматизация повторяющихся процедур может обеспечить реальную ценность для проектов разработки программного обеспечения. В этой статье мы рассмотрим значение и барьеры для автоматизации и предоставим некоторые рекомендации по автоматизации аспектов процесса разработки.

Хотя немногие опытные разработчики и руководители проектов будут спорить о преимуществах автоматизации процедур разработки и тестирования, когда толчок приходит к успеху, многие команды уделяют мало внимания внедрению автоматизированных процессов. Результатом обычно является то, что, если автоматизация вообще рассматривается, она предоставляется на словах в начале жизненного цикла проекта, но быстро отходит на второй план.

Опыт учит нас снова и снова, что попытка запустить «простой» проект путем реализации серии «простых» ручных протоколов, подкрепленных «простыми» письменными (иногда даже просто устными) инструкциями, просто не работает должным образом. Несмотря на это, многие из нас все еще склонны позволить себе начать следующий проект с мыслью, что ручной метод, основанный на протоколах, "подойдет".

В конце концов, не все ли мы профессионалы? Разве мы не можем прочитать набор простых инструкций и просто быть достаточно дисциплинированными, чтобы следовать этим инструкциям, когда придет время? Разве это не пустая трата времени и денег, чтобы инвестировать в автоматизацию процедур для такого маленького проекта? В конце концов, команда разработчиков - это всего полдюжины человек, и спор против автоматизации продолжается и продолжается.

Если вы разработчик или тестировщик, и вам нравится тратить свое время, фактически добавляя ценность своему проекту, вместо того, чтобы повторять одни и те же рутинные задачи снова и снова, вы захотите рассмотреть вопрос об использовании концепции автоматизации для вашей команды (особенно для вашего менеджера проектов). Если вы менеджер проекта, который стремится максимально использовать таланты и время членов вашей технической команды, а также минимизировать риск того, что ваш проект не будет выполнен вовремя и качественно, вы захотите призвать свою команду инвестировать необходимое время и усилия, необходимые для автоматизации типов задач, которые будут определены в этой статье.

Почему я должен автоматизировать?

Возможно, вы уже знакомы со многими преимуществами автоматизации процессов разработки. Вот некоторые из наиболее часто цитируемых:

Повторяемость. Сценарии могут повторяться, и, если у вашего компьютера особенно плохой день, вы можете быть достаточно уверены, что одни и те же инструкции будут выполняться в одном и том же порядке при каждом запуске одного и того же сценария.

Надежность. Скрипты уменьшают вероятность человеческой ошибки.

Эффективность. Автоматизированные задачи часто выполняются быстрее, чем те же задачи, выполняемые вручную. (Некоторые люди могут задаться вопросом, является ли повышение эффективности типичным, отмечая, что они работали над проектами, в которых, по их мнению, попытки автоматизировать задачи на самом деле стоят проекту больше времени, чем сэкономили. В зависимости от ситуации это может представлять реальную проблему. Кроме того, автоматизация, возможно, была плохо реализована или слишком далеко зашла в некоторых проектах, но продолжайте читать, чтобы узнать больше о том, что и когда нужно автоматизировать.)

Тестирование. Скриптовые процессы проходят тестирование на протяжении всего цикла разработки, почти так же, как системный код. Это значительно повышает шансы на успешное выполнение процесса по мере продвижения проекта. Автоматизированные сценарии в конечном итоге представляют собой зрелый, проверенный набор повторяющихся процессов.

Versioning. Скрипты - это артефакты, которые можно поместить под контроль версий. При ручных процессах единственными артефактами, которые могут быть версионированы и отслежены, являются документы процедур. Управление версиями людей - другой фактор в уравнении ручного процесса - к сожалению, не поддерживается вашей типичной системой управления источниками.

Кредитное плечо. Еще одним большим преимуществом автоматизации является то, что разработчики и тестировщики могут сосредоточиться на тех областях, где они добавляют реальную ценность проекту - разработке и тестированию нового кода и функций - вместо того, чтобы беспокоиться о проблемах, лежащих в основе инфраструктуры разработки.

Например, вместо того, чтобы требовать, чтобы все были близко знакомы со всеми мелкими нюансами процедуры сборки, вы можете попросить одного человека сосредоточиться на автоматизации сборки, и этот человек предоставит команде значительно упрощенный метод сборки, который, как мы надеемся, будет таким же простым, как запустить команду или два. Меньшее время, затрачиваемое на сборку, оставляет больше времени для задач, которые повышают ценность проекта.

Яндекс переводчик

Автоматизация повторяющихся процедур может обеспечить реальную ценность для проектов разработки программного обеспечения. В этой статье мы рассмотрим значение и барьеры для автоматизации и предоставим некоторые рекомендации по автоматизации аспектов процесса разработки.

Хотя немногие опытные разработчики и руководители проектов будут спорить о преимуществах автоматизации процедур разработки и тестирования, когда дело доходит до толчка, многие команды ставят низкий приоритет на внедрение автоматизированных процессов. В результате обычно, если автоматизация рассматривается вообще, она дается на словах в начале жизненного цикла проекта, но быстро падает на обочину.

Опыт учит нас снова и снова, что попытка запустить "простой" проект путем реализации серии "простых" ручных протоколов, подкрепленных "простыми" письменными (иногда даже просто устными) инструкциями, просто не работает хорошо. Тем не менее, многие из нас все еще склонны позволять себе начинать следующий проект с мыслью, что ручной, основанный на протоколе метод "отлично подойдет."

В конце концов, разве все мы не профессионалы? Разве мы не можем прочитать набор простых инструкций и просто быть достаточно дисциплинированными, чтобы следовать этим инструкциям, когда придет время? Разве это не пустая трата времени и денег, чтобы инвестировать в автоматизацию процедур для такого небольшого проекта? В конце концов, команда разработчиков—это всего лишь полдюжины человек—и аргумент против автоматизации продолжается и продолжается.

Если вы разработчик или тестировщик, которому нравится тратить свое время, фактически добавляя ценность своему проекту, а не повторяя одни и те же рутинные задачи снова и снова, вы захотите рассмотреть возможность пропаганды концепции автоматизации для своей команды (особенно для вашего менеджера проекта). Если вы являетесь менеджером проекта, который стремится максимизировать таланты и время членов вашей технической команды, а также минимизировать риск того, что ваш проект не сможет выполнить вовремя и качественно, вы захотите побудить свою команду инвестировать необходимое время и усилия, необходимые для автоматизации типов задач, которые будут определены в этой статье.

Почему Я Должен Автоматизировать?

Возможно, вы уже знакомы со многими преимуществами автоматизации процессов разработки. Некоторые из наиболее часто цитируемых из них:

Повторяемость. Сценарии могут повторяться, и, если у вашего компьютера нет особенно плохого дня, вы можете быть достаточно уверены, что одни и те же инструкции будут выполняться в одном и том же порядке каждый раз, когда запускается один и тот же сценарий.

Надежность. Сценарии уменьшают вероятность человеческой ошибки.

Эффективность. Автоматизированные задачи часто выполняются быстрее, чем те же задачи, выполняемые вручную. (Некоторые люди могут

задаваться вопросом, является ли повышение эффективности типичным, отмечая, что они работали над проектами, где, по их мнению, попытка автоматизировать задачи фактически стоила проекту больше времени, чем он сэкономил. В зависимости от ситуации, это может быть реальной проблемой. Кроме того, автоматизация может быть реализована плохо или слишком далеко на некоторых проектах—но продолжайте читать больше о том, что автоматизировать и когда.)

Тестирование. Скриптовые процессы проходят тестирование на протяжении всего цикла разработки, во многом так же, как и системный код. Это значительно повышает шансы на успешное выполнение процесса по мере развития проекта. Автоматизированные сценарии в конечном итоге представляют собой зрелый, проверенный набор повторяемых процессов.

Управление версиями. Скрипты-это артефакты, которые можно поместить под управление версиями. При ручных процессах единственными артефактами, которые могут быть версированы и отслежены, являются документы процедур. Управление версиями человека-другой фактор в уравнении ручного процесса-к сожалению, не поддерживается вашей типичной системой управления версиями.

Рычаг. Еще одно большое преимущество автоматизации заключается в том, что разработчики и тестировщики могут сосредоточиться на тех областях, где они добавляют реальную ценность проекту—разработке и тестировании нового кода и функций—вместо того, чтобы беспокоиться о базовых проблемах инфраструктуры разработки.

Например, вместо того чтобы требовать, чтобы все были близко знакомы со всеми маленькими нюансами процедуры сборки, вы можете попросить одного человека сосредоточиться на автоматизации сборки и предоставить команде значительно упрощенный метод построения, надеюсь, такой же простой, как выполнение команды или двух. Меньшее время, затрачиваемое на построение, оставляет больше времени для задач, которые добавляют наибольшую ценность проекту.

4. Выпишите незнакомые слова и определите их переводы с помощью <https://www.lingvolive.com>

| | |
|-------------------------|---|
| <div>repetitive</div> | <div> <div>The Universal Dictionary ⓘ</div> <div> <div>repetitive</div> <div>[rɪˈpetətɪv]</div> <div>= repetitious</div> </div> </div> <hr/> <div> <div>Physics ⓘ</div> <div> <div>repetitive</div> <div>периодически повторяющийся, повторяющийся</div> </div> </div> |
| <div>implementing</div> | <div> <div>User translations</div> <div> <div>_____ The part of speech is not specified _____</div> <div>1. ВОПЛОЩАЯ</div> <div>translation added by Elena Abashkina ⭐ BRONZE EN-RU</div> </div> </div> |
| <div>Efficiency</div> | <div> <div>The Universal Dictionary ⓘ</div> <div> <div>efficiency</div> <div>[ɪˈfɪ(ə)n(t)si] , [ə-] брит. ⓘ / амер. ⓘ</div> <div>сущ.</div> <div> <div>1) эффективность, результативность, действенность</div> <div>2) продуктивность, производительность</div> <div>3) умение, деловитость, оперативность</div> <div>4) тех. отдача, коэффициент полезного действия</div> <div>5) амер. маленькая квартира с минимальными кухонными и сантехническими удобствами</div> </div> </div> <hr/> <div> <div>Learner's Dictionary ⓘ</div> <div> <div>efficiency</div> <div>[ɪˈfɪənsɪ] ⓘ</div> <div>н</div> <div>действенность, эффективность</div> </div> </div> </div> |
| <div>Leverage</div> | <div> <div>The Universal Dictionary ⓘ</div> <div> <div>leverage</div> <div>[ˈliːv(ə)rɪʒ] брит. ⓘ / амер. ⓘ</div> <div>1. сущ.</div> <div> <div>1) действие рычага; система рычагов</div> <div>2) сила, усилие рычага</div> <div>3) средство для достижения цели</div> <div>4) фин. использование кредита для совершения финансовых сделок</div> </div> <div>2. гл.</div> <div> <div>1) использовать (с выгодой для себя)</div> <div>2) фин. использовать кредит для совершения финансовых сделок</div> </div> </div> <hr/> <div> <div>Patents and Trademarks ⓘ</div> <div> <div>leverage</div> <div>средство для достижения чего-л.</div> </div> </div> </div> |

