

by Michael DeHaan, a developer who has been working with configuration management and infrastructure orchestration in one form or another for many years. He has worked with Puppet, Chef, Cfengine, server deployment (Capistrano, Fabric,) and ad-hoc task execution (Func, plain SSH), and wanted to see if there was a better way. Ansible wraps up all three of these features into one tool, and does it in a way that's actually simpler and more consistent than any of the other task-specific tools.

Ansible is a automation and provisioning tool that makes it easy to configure systems with the needed software, configuration options and even content. It is a command line tool, written in Python, that uses SSH connections to run these actions. This means that all you need to do is have a SSH connection to a machine and Ansible will run any actions you want to run.

Using Ansible it's easy to deploy — and most importantly, it uses a very simple language (YAML, in the form of Ansible Playbooks) that allow you to describe your automation jobs in a way that approaches plain English.

Cfengine, Chef and Puppet are fantastic and can be used to manage extremely large infrastructures but there is no denying that they have a large learning curve and can be difficult to setup and configure. Ansible aims to be simpler and easier to understand while still maintaining the efficiency and power of others tools.

#### **References:**

- Jeff Geerling , 2014, “Ansible for DevOps ”
- <http://docs.ansible.com/>

### ***Програмне забезпечення ІТ-компанії***

*Скоропад О.*

*Компанія EPAM, sko@ukr.net*

Software development company is a factory for the production of IT products. Like any factory, this company has complex structure and big set of processes with a lot of the necessary tools. All these tools are special applications for software development. This article describes a simple software landscape of modern IT company and explains major functionality of components. Particular attention is paid to free software.

Сучасна ІТ компанія – це фабрика з випуску програмних продуктів. Як і кожна фабрика, така компанія має свою структуру та налагоджений складний виробничний процес з великою кількістю необхідних інструментів. В ролі інструментів виступають програмні продукти призначені для розробки на всіх його етапах програмного забезпечення, а оскільки таких етапів є багато, то і перелік продуктів є дуже широким.

Спробуємо класифікувати програмне забезпечення для розробки:

1. Інструменти програмної архітектури та аналітики призначені для візуалізації ідеї програмного проекту та візуалізації блок-схеми та основних

компонентів майбутнього ПЗ;

2. Інструменти розробника (редактори, бази знань та інструменти відслідковування помилок /bug-track tool/ );
3. Засоби контролю коду (code review);
4. Система контролю версій;
5. Центральний репозитарій програмного коду;
6. Програмні засоби безперервної інтеграції;
7. Програмні засоби автоматичного та ручного тестування;
8. Засоби доставки продукту до користувача, а також засоби комунікації з клієнтом.

Окремої уваги вимагає системне програмне забезпечення для підтримки роботи цієї інфраструктури, яке проте не входить в цей огляд:

1. Засоби віртуалізації;
2. Програмне забезпечення автоматизованої конфігурації;
- 3 Засоби розгортання програмних середовищ (orchestration tool) та інструменти управління контейнерами;
4. Засоби моніторингу;
5. Програмне забезпечення для резервного копіювання.

Це основний, але далеко не повний перелік типів продуктів, який розширюється настільки швидко, наскільки відбувається все вужча спеціалізація в індустрії програмного забезпечення.

Варто зупинитись на природі перерахованих продуктів. Якщо раніше і розробники програмного забезпечення і його користувачі віддавали перевагу дорогим брендовим продуктам відомих компаній за принципом дорожче – краще, то зараз ситуація на IT-ринку кардинально змінилась: перевага надається безоплатно поширюваному, умовно-платному, а найбільше - відкритому програмному забезпеченню. Така переорієнтація ринку відбулась з наступних причин:

1. Код відкритого ПЗ вільний для модифікації чим забезпечується максимальна гнучкість проектів;
2. Відкрите програмне забезпечення розвивається, по суті під наглядом спільноти, тому суттєво знижуються ризики пов'язані з неякісним кодом;
3. Повністю ліквідуються загрози, пов'язані з припиненням розробником супроводу програмного продукту, оскільки супровід відбувається консолідованими зусиллями самих користувачів;
4. Вартість проектів з використанням відкритого ПЗ на порядок нижча за вартість таких же проектів з використанням комерційного ПЗ.

Всі ці фактори приводять до того, що не лише користувачі, а й дедалі більше розробників починають переходити на відкрите ПЗ. Великі компанії з розробки ПЗ ефективно використовують практику тестування нових версій продуктів в якості відкритого ПЗ, а лише після цього ліцензують код найбільш стабільних версій. Яскравим прикладом такого методу є тандем CentOS – RedHat.

Проведемо короткий огляд основних спеціальних програмних продуктів, що широко використовуються компаніями з розробки ПЗ в промисловій експлуатації:

1. Засоби контролю коду – це програмне забезпечення, яке призначене для первинної перевірки програмного коду. Інша важлива функція цього ПЗ полягає в передачі знань від однієї людини, що пише код до інших спеціалістів, які в разі потреби зможуть продовжити написання програми. Вказане ПЗ часто вбудовується в системи контролю версій. Популярні пакети: Gerrit, Barkeeper, Phabricator;

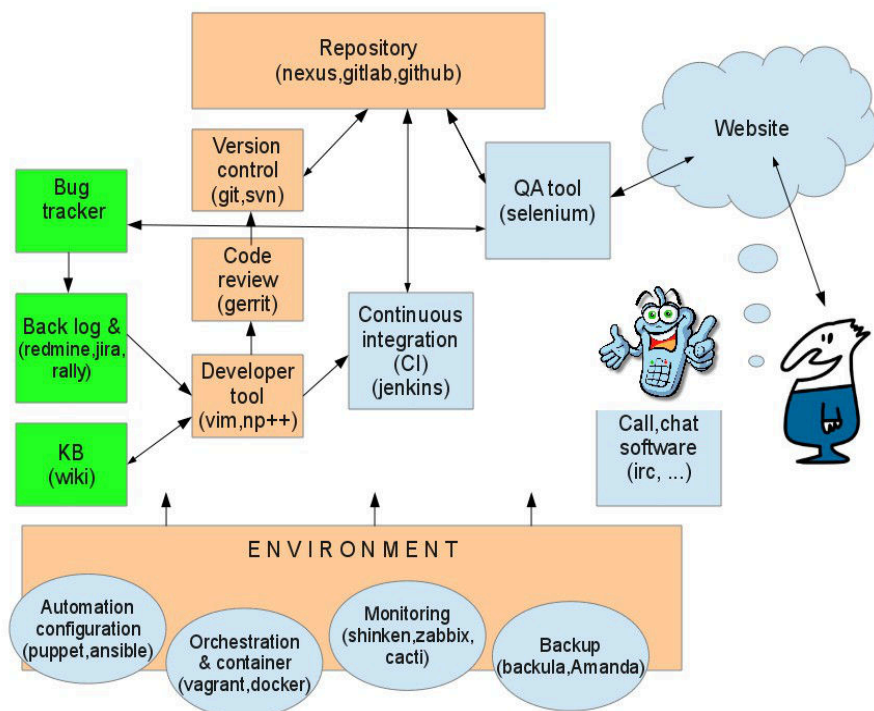
2. Системи контролю версій та репозитар – це централізоване сховище коду з організованою структуризацією версій ПЗ. Є головною базою всіх напрацьованих продуктів, дає можливість повернення в будь-який момент до будь-якої зафіксованої версії ПЗ, а також дає змогу створити незалежну гілку написання програмного продукту з довільної версії поточного продукту. Найвідоміші продукти: svn, git, gitlab, nexus;

3. Системи безперервної інтеграції (CI - continuous integration) з'явилися відносно недавно у зв'язку з переходом на нові технології розробки ПЗ – Agile та Scrum. Це ПЗ дає змогу повністю автоматизувати найбільш рутинні операції розробника, якими є : завантаження залежностей та бібліотек, компіляція, блок-тестування, публікація результатів. Системи CI практично завжди працюють в тандемі з системою контролю версій та системами тестування, що дає змогу, наприклад, здійснювати автоматичне повернення до попередньої версії проєктованого ПЗ за наявності помилок в його коді. Приклади пакетів: Jenkins, Metacube (комерційне ПЗ), RHPCI;

4. Засоби тестування – широка лінійка програмних продуктів, що працює в галузі забезпечення надійності програмного коду (QA – quality assurance). Вказане ПЗ призначене для контролю якості робочих версій коду і дає змогу відсіяти у разі хорошої організації процесу до 90% помилок в продуктах. Галузь QA інтенсивно розвивається, що приводить до появи великої кількості нового ПЗ для тестування. На сьогодні одних лише типів тестів нараховується близько десятка і для кожного з них існує своя лінійка тестового ПЗ. Найвідоміші пакети: Selenium IDE, Sikuli, WebDriver, PHPUnit, Amoc.

Як вже було зазначено, одною з найбільших переваг відкритого програмного забезпечення є широка доступність документації та навчальних матеріалів, що дає змогу ефективно та максимально швидко готувати персонал для роботи з ним. Процес підготовки проходить у вигляді так званих техконференцій у максимально вільному форматі, де встановлюється лише загальна тема та перелік властивостей програмного забезпечення, що обов'язкові до розгляду. Такий вид навчання нагадує обмін знаннями в професійних спільнотах і дуже добре зарекомендував себе.

На рис. 1 зображена блок-схема базового ПЗ, що широко використовується розробниками програмного забезпечення.



***Навчальний посібник з використання програми SCRIBUS для  
початківців та фахівців  
Хамула О.Г., Дмитрів Л.Й.***

Українська академія друкарства, кафедра видавничої справи і редагування,  
кафедра інформаційних мультимедійних технологій [khamula@gmail.com](mailto:khamula@gmail.com)  
[lidmytriv@gmail.com](mailto:lidmytriv@gmail.com)

The article presents a textbook about Scribus – modern software in publishing business for beginners and specialists. Also the article gives the marketing research results about free software in the field.

Перехід до використання у видавничій справі вільного програмного забезпечення під керуванням різних операційних систем, зокрема на ядрі Linux, зумовив появу Scribus – найпотужнішого у даний час крос-платформного продукту для автоматизації видавничого процесу, що поширюється за умовами ліцензії GNU (General Public License) та підтримує кольорові моделі СМΥК та RGB. Програма дає змогу на професійному рівні вирішувати найширший спектр завдань з верстання як друкованих, так і