

# УСТАВ ДЛЯ ПРОГРАММИСТА

## 0. Развертывание Друпала - базовые действия

- Развернуть и установить Drupal 7 последней версии
- В установленном Drupal не должно быть файлов install.php, \*.txt (кроме robots.txt)
- Положить под GIT с настроенными ветками (подробно см. Пункт 1).
- Настроить файл robots.txt (подробно см. Пункт 2).
- Логин / пароль администратора сайта (он же супер-админ, он же пользователь с id = 1) должен быть такой же как и логин / пароль к базе данных сайта. В случае изменения логина / пароля к базе данных- логин / пароль супер-администратора должен измениться соответственно. Логин / пароль супер-администратора можно найти во вкладке "Info" проекта, если его там нет, сообщите DevLead / менеджера об этом.
- Установить базовые модули (подробно см. Пункт 3).
- Выключить ненужные модули.
- Настроить файловую систему (подробно см. Пункт 11.2).
- Проверить статус репорт на наличие проблем.
- По предварительной договоренности с заказчиком и при необходимости сайт можно положить под .htpasswd.

## 1. GIT

- 1.1 Каждый сайт должен быть под GIT-ом;
- 1.2 Все на сайте должно лежать под GIT-ом (кроме отведенной файловой папки)
- 1.3 Все изменения на живом и / или dev сайте должны быть закомичены;
- 1.4 В GIT репозитории ветвь master должна содержать файл README с названием проекта;
- 1.5 Также должна быть ветка с названием dev - здесь собственно и должен быть сайт;
- 1.6 После выполнения задачи, если есть необходимость, всегда комитить и пушить изменения;
- 1.7 В конце рабочего дня на проекте в статусе GIT не должно быть незакомиченных изменений. Если те изменения в файлах, которые не были закомичены, магическим образом исчезнут на следующий день - ваши проблемы.
- 1.8 Писать адекватные и развернутые комментарии к комитам.
- 1.9 Если у какого-то пользователя нет доступа к репозиторию (а он должен быть), то Development Lead должен сбросить СТО в скайп в частную ветку переписки:
  - Ник пользователя;
  - Ссылку на репозиторий в Лабе.
- 1.10 Ответственный за то, чтобы адекватно велся контроль версий - Development Lead.
- 1.11 Если нужно выдать доступ к нашему GIT репозиторию заказчику, то DevLead должен:
  - а) получить апрув от менеджера о перманентном доступе заказчика к актуальной версии нашего кода

- б) смириться с мыслью, что наш гавнокод будет видеть заказчик;
- в) сбросить СТО в скайп в частную ветку переписки:
  - Email заказчика (важно: этот Email должен соответствовать Email-у, что будет указан в параметре "автор" комита)
  - Публичный ключ заказчика
  - Ссылки на репозиторий в Лабе.
- 1.12 Ветки никогда не удаляются (право на удаление может предоставить только DL проекта в исключительно индивидуальных случаях).
- 1.13 Ядро Drupal нужно устанавливать в качестве сабмодуля GIT-а. Рекомендации по установлению можно найти в блоге [Drupal Core Installation As A Submodule](#), репозиторий с Drupal-ом находится здесь - [Drupal core](#)
- 1.14 Описание Комит (Commit messages)
  - 1.14.1 ВАЖНО: не начинайте описание комита со знака #. Это может привести к значительным неудобствам при работе с вашим комитом, в первую очередь при использовании rebase -i.
  - 1.14.2 Описание комита пишется на английском языке.
  - 1.14.3 Сообщение комита должно выглядеть следующим образом: Task # [номер задачи]: [краткое описание изменений].
  - 1.14.4 Описание комита должен начинаться с глагола в прошедшем времени, в соответствии с типом задачи и характера изменений:
    - Added (новый функционал, New Feature)
    - Fixed (исправлена ошибка, Bug)
    - Changed (изменение существующего функционала, Task)
    - Updated (изменение кода для соответствия изменениям сторонней библиотеки или API, Task)

### Общее по работе "dev-live site"

Предусмотрены такие основные branch-и:

- Live - ветвь живого сайта;
- Dev - ветвь дев сайта;
- Одна ветвь для каждого тикета.

Начало работы с задачей:

- `git checkout live` - переключиться на лайв бранч.
- `git checkout -b [ticket: id]` - создать и сразу переключиться в новый бранч задачи.

**Примечания:**

- Название бранчи в GIT напрямую зависит от номера задачи, например, если номер задачи 150, тогда бранч нужно называть так: "150" (без кавычек).

Конец работы с задачей:

- `git checkout [ticket: id]` - переключиться на бранч, который вы завершили.
- `git push origin [ticket: id]` - запустить бранч с задачей (конечно, все должно быть закомичено).
- `git pull origin live` - подтянуть все изменения на live.
- `git merge live` - змерджить все изменения с live в бранч тикета.
- `git checkout dev` - переключиться на dev ветку.
- `git pull origin dev` - запустить dev ветку.
- `git merge [ticket: id]` - смержить изменения ветки тикета в dev бранч.
- `git push origin dev` - запустить dev бранч.

**Примечания:**

- Не забывать об обновлении сабмодулей (`git submodule init && git submodule update`).

- Перед пушем в dev бранч стоит проверить работоспособность сайта.

**ВАЖНО:**

- Никто не комитит в dev бранч, туда только мержить тикеты.
- Никто не комитит и не мержит в live бранч (под исключения подпадают непредсказуемые обстоятельства, ответственность за которые несет DL проекта).
- После переноса (изменение live ветви) нужно смержить рабочие бранчи с актуализированной (!) Live ветвью.
- Не стоит делать 2 комита в 1 бранч. Исключением может послужить тот случай, если вы задачу выполняете в течение 2 дней или больше. Но пытаться делать только 1 комит (это служит для надежности мержа тикетов).

Любые правила по работе "dev-live site" могут быть изменены, но не исключены DL проекта.

## 2. robots.txt

2.1 Сайт не должен индексироваться поисковыми системами, если он находится на dev сайте (на drupalcompany.com все уже настроено, поэтому изменять файл не нужно, только проверить);

2.2 Файл должен находиться под GIT-игнором.

2.3 По умолчанию live сайт должен быть открыт для индексирования поисковыми системами.

## 3. Установка модулей

3.1 все модули должны быть установлены в качестве GIT сабмодулей;

3.2 если необходимый вам модуль нужно ставить в качестве сабмодуля (например `podejs`), устанавливая такой модуль разрешается только с разрешения старшего программиста проекта;

3.3 для базовой сборки обязательно инсталляция и настройка следующих модулей / плагинов:

- GlobalRedirect
- Administration menu
- Views
- Pathauto
- GoogleAnalytics
- XML Sitemap
- Elysia Cron
- Metatags
- WYSIWYG
- IMCEm2
- редактор TinyMCE (по умолчанию, если нужен другой текстовый редактор, ставим другой)
- WYSIWYG filter
- Transliteration
- Subpathauto

3.4 Не стоит ставить все модули, имеющиеся на Drupal.org. Если вам необходимо к сайту задействовать незначительный функционал и есть возможность написать этот функционал в собственном модуле, - обратитесь за рекомендацией к старшему программисту проекта.

3.5 Все модули, которые вы не знаете, можно устанавливать только после разрешения старшего программиста проекта.

#### 4. Тема и Темизация

4.1 Тема, которую необходимо установить, должна согласовываться со старшим программистом и с верстальщиком, или менеджером проекта. По умолчанию ставить тему Ninesixty;

4.2 Все tpl файлы (\* .tpl.php) должны находиться в папке templates;

4.3 Все tpl файлы, которые находятся в папке templates, в свою очередь должны находиться в папках по модулям. Например, если у нас есть views-view.tpl.php, тогда этот файл должен быть в папке THEME / templates / views / views-view.tpl.php.

#### 5. Процесс работы над сайтом

5.1 Ни в коем случае не "хачить" модули и ядро Drupal;

5.2 Функционал должен нести максимально динамичный характер (то есть максимально избегать проблем с переписыванием функционала, потому что был плохо продуман);

5.3 Написанный код должен быть понятным другим программистам (желательно писать комментарии), давать адекватные названия модуля и его описание, избегать создания переменных типа \$query1 \$query2 или \$a \$b \$c;

5.4 Соблюдать соответствия размещения кода и целостности его функциональности. (Например, не допускать следующих вещей: когда исключаем кастомные модули - сайт не работает или часть функционала, которая относится к модулю, еще до сих пор в работе.)

5.5 Разработку сайта вести на локальном сайте (развернутом на вашем компьютере, на компьютере старшего программиста или программиста)

5.6 На сайте не должны быть включены модули, которые не принимают участия в работе сайта;

5.7 В папке с загруженными файлами (sites / default / files) держать порядок, не сбрасывать все загруженные файлы в единую папку.

- 5.8 Запрещается писать кириллицу в коде.
- 5.9 Код должен соответствовать coding-стандартам (php / js / css / comments).
- 5.10 Все резолвы в проекте должны идти через Dev Lead-a.
- 5.11 За все задачи в проекте отвечает Development Lead (верстка, тестирование, программирование, администрирование).

## 6. Правила именования и размещения кода

### 6.1 Custom модули собственного производства

6.1.1 Основной модуль (или если он единственный) должен находиться в директории: "sites / all / modules / custom / internetdevels.com". Name в info файле - "idevels".

6.1.2 Другие custom-модули должны называться по паттерну idevels\_ \* и размещаться в директории: "sites / all / modules / custom / % module\_name%".

6.1.4 Исключением относительно такого наименования или размещения модулей могут быть только те случаи, где DL, менеджер или СТО прямым текстом это указали.

6.2 Contrib модули (модули с Drupal.org) должны находиться в директории: "Sites / all / modules / contrib / % module\_name%".

6.3 Модули типа "feature" должны находиться в директории: "Sites / all / modules / features / % feature\_name%".

6.4 Пропатченные контриб модули должны лежать в папке sites / all / modules / modified / % module\_name% вместе с патчем. Пропатченный модуль удаляется с сабмодулей и устанавливается как кастомный, но в вышеупомянутую папку.

6.5 Основная тема должна находиться в директории "sites / all / themes" и иметь однозвучное с названием проекта наименование.

## 7. Что делать, если сайт не адекватно работает или вообще не работает

7.1 При возникновении проблем с сайтом нужно вспомнить, после чего эти проблемы начались, попытаться вернуть те изменения на сайте, которые были применены до возникновения проблемы.

7.2 Запомните, что все работы по восстановлению работы сайта требуется проводить на локальном сайте.

7.3 Если вы нашли баг в модуле / ядре нужно:

1) воспользоваться Google, поискать патч, или решение проблемы для данной версии модуля. Если найден необходимый патч, тогда читать пункт 5), если нет, читаем следующий пункт.

2) попытаться установить другую версию модуля (более новую или старую), если не помогло, читаем следующий пункт.

3) разобраться максимально подробно в чем проблема, подебажить (можно использовать xdebug-ом), убедиться не создает ли эту ошибку версия ядра сайта. Если версия ядра сайта устарела, попробовать установить более новую версию. Если не помогло, читаем следующий пункт.

4) если вы уверены в том, что можно самому написать патч, пишем, и тестируем. Если патч работает, тогда читать пункт 6.4.

5) в противном случае обратиться за помощью к Dev Lead-у проекта, или к коллегам по работе, которые имели дело с подобными вещами.

5) перед установкой патча обязательно нужно сообщить об этом Dev Lead-у и менеджеру проекта. Устанавливать патч только после апрува и предыдущих рекомендаций (если есть) этих лиц.

7.4 Не допускать безрезультатно "залипания" над проблемой! Пользуйтесь помощью Dev Lead-а или коллег по работе.

7.5 Если причина ошибки возникла не по нашей вине, и вы ее исправили, обязательно об этом необходимо сообщить Dev Lead-у и менеджеру проекта.

## 8. Доступы.

8.1 Ни в коем случае без разрешения Dev Lead-а и менеджера не предоставлять доступы заказчику до глобального пользователя (uid == 1).

8.2 Как правило, заказчику создается роль редактора, где он сможет заполнять сайт и / или руководствоваться его функциональными частями. Все доступы нужно согласовывать лично со старшим программистом / менеджером / заказчиком.

8.3 Не предавать заказчику лишних доступов к функциональным частям сайта.

8.4 Если вам необходимо дать один из доступов и этот доступ позволяет еще руководствоваться другим функционалом сайта, к которому заказчик не должен иметь доступ, нужно:

а) Предупредить заказчика об этом, рассказать о возможных последствиях неосознанного пользования тех функциональных частей сайта, которые Drupal физически не может ограничить.

б) Предложить заказчику сделать уникальные настройки доступов к функциональным частям сайта ("защита от дурака").

8.5 Не предавать заказчику доступы к Full HTML без предварительного согласования этого действия с Dev Lead и менеджером проекта. Без согласования, текст обязательно должен проходить фильтрацию.

## 9. Живой сайт

9.1 На живом сайте должен присутствовать и быть настроенным модуль Google Analytics.

9.2 При возможности должны быть настроенный модуль boost или кеширование в Drupal, включенное объединение CSS и JS.

9.3 Зависимо от типа проекта (кроме NDA) должен быть добавлен наш копирайт согласно правилам [SEO-методички](#) пункт "Авторская ссылка". Если есть какие-то сомнения или нужны уточнения, обращаться к CEO.

*Пункт от 24 марта 2014 года.*

## 10. Development Lead:

10.1 Отвечает за ВСЕ вопросы технического и архитектурного характера.

10.2 Отвечает за наличие и адекватную настройку всех обязательных модулей на сайте.

10.3 Отвечает за работу всех модулей, которые используются на сайте: их версии, релизы, совместимость с другими модулями и т.д.

10.4 Делает оценку (aka estimate) новых задач, обращаясь, если необходимо, к коллегам (верстальщикам или программистам в зависимости от своей позиции). За адекватность всех оценок полностью отвечает Development Lead (даже если не он её не делал).

10.5 Не отвечает за приоритетность задач, это обязанность менеджера проекта.

10.6 Не выставляет эффективные часы в проекте - это делает менеджер проекта, который может обратиться за консультацией по этому вопросу к PM lead.

10.7 На проектах типа Dedicated должен до 3-го числа каждого месяца предоставить менеджеру распределение заранее утверждённого бюджета между

участниками проекта.

Перенесено с "Положения для Development Lead" 2 сентября 2013 года.

## 11. Приложения:

### 11.1 Настройка кнопок для текстового редактора

В настройках должны быть включены следующие кнопки:

- Работа с шрифтом: Bold, Italic, Underline, Strike-through.
- Ссылки: Link, Unlink.
- Работа с текстом: Align right, Align left, Justify, Align center, Bullet list, Numbered

list.

- Дополнительные кнопки: Image (с поддержкой IMCE), Cut, Copy, Paste, Paste from Word, Paste text.

Включение других кнопок, или отключение существующих, обсуждается со старшим программистом / менеджером / заказчиком.

### 11.2 Настройки файловой системы

- tmp - должна находиться за путём sites/default/files/tmp;
- sites/default/files и все её внутренние папки должны иметь права 777;
- imce профили - профиль обычного пользователя должен иметь собственную папку без возможности выйти за пределы этой папки.

### 11.3 sites/default/files/

- Выгрузку файлов нужно настраивать по папкам. Для примера: у нас есть поле field\_image и тип контента acticle, соответственно все файлы загружаемые к материалам этого типа контента нужно сохранять в отдельной папке.

- Название папки должно иметь машинное название типа контента и название поля. Если файлов в этом поле будет очень много, тогда разделять папку на подпапки, например по дате.

- Имя файла по возможности делать в формате: [file-fid].[original-extension].

### 11.4 Алиасы

- На каждом сайте по умолчанию должен быть настроен модуль Pathauto.
- Алиасы задавать соответственно к разделам. Например, если у нас есть список новостей с ссылкой news, тогда каждая новость должна иметь алиас news/test-news.
- Алиасы должны быть транслитерированные.

### 11.5 Тип контенту

- Машина, название типа контента у Drupal должно быть описано одним словом.
- Одни филды можно/нужно юзать в разных типах контенту ( при возможности нужно юзать один филд во многих типах контента, чтобы избежать ситуаций, когда на сайте есть филды "field\_image", "field\_news\_image", "field\_blog\_image" с одинаковыми настройками).

Пункт от 24 июля 2013 года.

## 12. Контент:

12.1 Тестовый контент (изображение, видео, текст) должен максимально отвечать тематике сайта и быть адекватным.

12.2 В качестве тестового текста можно использовать [Lorem ipsum](#).

12.3 Запрещается использовать политическую или религиозную тематику.

