Best Practices & (Coding) Standards

Johannes Haseitl

@derhasi @undpaul

Überblick

1. Warum sind Standards wichtig
2. Beispiele: Coding, Drupal, Tools
3. Standards einführen
4. Q&A

Warum sind (Coding) Standards wichtig?

ein eindeutige "Lösung"

- Abnahme von trivialen Entscheidungen
- Konzentration auf relevante Dinge

Uniformität

- bessere Wiedererkennung
- bessere Übersicht
- bessere Unterscheidung

Jeder arbeitet immer im Team

- Kunde
- **Future Me**

Information

- mehr Informationen
- => bessere/nachhaltigere Entscheidungen
- WAS, WANN, WO, **WARUM** ?

Best practice #1

VCS benutzen

- wer arbeitet nicht mit Git?
- Nachvollziehbarkeit, "git blame"
- wer hat WAS, WANN, WARUM gemacht?
- => mehr Information

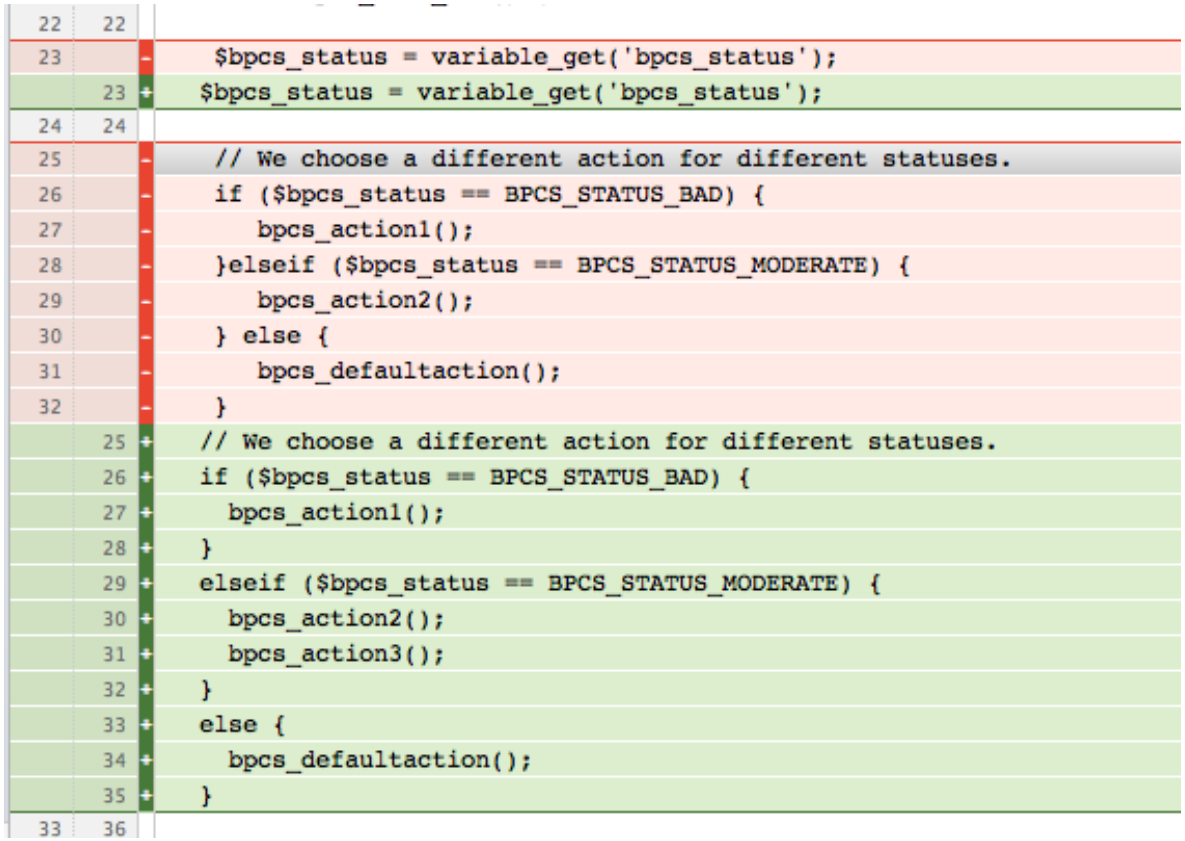
Coding standards

Drupal Coding standards

- https://drupal.org/coding-standards
- Coder
- auch für CSS und Javascript

```
if (condition1 || condition2) {
  action1;
}
elseif (condition3 && condition4) {
  action2;
}
else {
  defaultaction;
}
```

Diff



```
22  22
23      -   $bpcs_status = variable_get('bpcs_status');
    23  +  $bpcs_status = variable_get('bpcs_status');
24  24
25      -   // We choose a different action for different statuses.
26      -   if ($bpcs_status == BPCS_STATUS_BAD) {
27      -      bpcs_action1();
28      -   }elseif ($bpcs_status == BPCS_STATUS_MODERATE) {
29      -      bpcs_action2();
30      -   } else {
31      -      bpcs_defaultaction();
32      -   }
    25  +  // We choose a different action for different statuses.
    26  +  if ($bpcs_status == BPCS_STATUS_BAD) {
    27  +    bpcs_action1();
    28  +  }
    29  +  elseif ($bpcs_status == BPCS_STATUS_MODERATE) {
    30  +    bpcs_action2();
    31  +    bpcs_action3();
    32  +  }
    33  +  else {
    34  +    bpcs_defaultaction();
    35  +  }
33  36
```

- funktionelle Änderung?
- Anpassung von Coding Standards in seperatem Commit

Guten Code schreiben

- kein Spaghetti-Code
- Nesting minimieren
- Funktionalität kapseln

Sinnvolle Benennung

- `$ln` vs `$line_number`
- `renderLink()` vs. `buildLink()`
- Form submit / validate

```
$form['actions']['revert'] = array(
  '#type' => 'submit',
  '#submit' => array(
    'myfancymodule_settings_form_submit_revert',
  ),
);
```

Code-Dokumentation

Jede Funktion kommentieren

Jeden Parameter und Rückgabewert kommentieren

Jeden Typen deklarieren

- `@param string $machine_name`
- `@return mixed?`
- `@param string | object?`

Logik beschreiben

- Was ist das Ziel?
- Warum mach ich das?

- @file
- @param
- @return
- @see
- @todo

Error reporting

- `error_reporting = E_ALL | E_STRICT`
- Warnings können strukturelle/logische Fehler aufdecken.

Larry Garfield

- https://london2011.drupal.org/conference/sessions/code-stinks
- https://munich2012.drupal.org/program/sessions/functional-php.html
- https://prague2013.drupal.org/session/aphorisms-api-design

Drupal

Benennung von Views, Panels und Co.

- machine_name = Name
- Beschreibung nutzen
- Sinnvolle Tags: z.B. `news`, `admin`
- Tags im Team vereinbaren
- Namespacing: `news_teaser`, `news_links`

Views

- Nicht zu viele Display Varianten
- Vererbung zu komplex: "Aus Versehen Feld geändert"
- "Views Content Panes" statt "Block" benutzen
- "Embed display" für Code verwenden

Module-Struktur

- contrib
- custom
  - sandbox: SOURCE.md
- features

Contrib-Patches

- **Don't hack core!**
- Datei im Contrib-Modul ablegen
- => `drush dl` & `git diff`
- Issue auf drupal.org ist Pflicht
- Patch-Datei trägt Issue-Nummer

Features

- nicht alle Inhaltstypen in ein Feature
- Feature = Bundle einer abgeschlossenen Funktion/Zusatzfunktion
- Dependencies klein halten: u.a. Permissions in das richtige Feature

Tools

Git

- Sinnvolle Git-Commits
- auch für CSS Anpassungen
- Git-Commits mit Issue Nr.
- **git nutzen!!**
  - **`git diff`** => review vor commit/push
  - **`git cherry-pick`**

Automatisieren

- "Human factor" minimieren
- update.sh
- Jenkins

Standards im Projekt

Kommunikation zentralisieren

- zentrales Tool: Redmine, ERPAL, Jira, OpenAtrium, ...
- Zentraler Ort für Dokumentation

Spielregeln

- Akzeptanzkriterien
- ReadyForTest
- Definition of Done

Standards einführen

Konsens

zentral festhalten

Einhaltung kontrollieren

Q / A

Danke!

Mehr in unserem undpaul Drupal Blog!