Welcome

The Seven Steps To Better PHP Code

(Part Two)

Stefan Priebsch, e-novative GmbH

Who I Am

- PHP enthusiast since 2000
- IT and PHP Consultant
- From Munich, Germany
- University Degree in Computer Science
- Writer (Books and Articles)
- Blog: *http://inside.e-novative.de*
- Email: *stefan.priebsch@e-novative.de*

What is Refactoring?

- Changing code
  - to improve readability
  - to simplify the structure
- *Without* changing the results
- *Not* adding new functionality

Shameless Plug ;-)

- Zend Studio for Eclipse has Refactoring Support
  - Move files and folders
  - Rename files, classes, functions, and variables
  - Organize includes
- http://www.zend.com/en/products/studio/

Where we left off

1. Format Source Code consistently
2. Use Consistent Naming
3. Document the API

For details, see part 1 of this Webinar at

http://www.zend.com/resources/webinars

Eliminate Redundant Code

- Avoid duplicate or similar code
- Goal: Make code changes in just one spot
- Create parametrized functions and methods („helper functions“)
- Do not over-generalize when programming, use Refactoring to

```
if (isset($_REQUEST['action']))
{
  if ('add' == $_REQUEST['action'])
  {
    $q = $db->query('SELECT quantity FROM data WHERE item=\'' . $_REQUEST['item'] . '\'');
    $quantity = $q->current(SQLITE_NUM);
    $quantity = $quantity[0];

    $quantity++;

    $q = $db->queryExec('UPDATE data SET quantity=' . $quantity . ' WHERE item=\'' .
         $_REQUEST['item'] . '\'');
  }

  if ('remove' == $_REQUEST['action'])
  {
    $q = $db->query('SELECT quantity FROM data WHERE item=\'' . $_REQUEST['item'] . '\'');
    $quantity = $q->current(SQLITE_NUM);
    $quantity = $quantity[0];

    if ($quantity > 0) $quantity--;

    $q = $db->queryExec('UPDATE data SET quantity=' . $quantity . ' WHERE item=\'' .
         $_REQUEST['item'] . '\'');
  }
}
```

```
if (isset($_REQUEST['action']))
{
  if ('add' == $_REQUEST['action'])
  {
    $q = $db->query('SELECT quantity FROM data WHERE item=\'' . $_REQUEST['item'] . '\'');
    $quantity = $q->current(SQLITE_NUM);
    $quantity = $quantity[0];

    $quantity++;

    $q = $db->queryExec('UPDATE data SET quantity=' . $quantity . ' WHERE item=\'' .
         $_REQUEST['item'] . '\'');
  }

  if ('remove' == $_REQUEST['action'])
  {
    $q = $db->query('SELECT quantity FROM data WHERE item=\'' . $_REQUEST['item'] . '\'');
    $quantity = $q->current(SQLITE_NUM);
    $quantity = $quantity[0];

    if ($quantity > 0) $quantity--;

    $q = $db->queryExec('UPDATE data SET quantity=' . $quantity . ' WHERE item=\'' .
         $_REQUEST['item'] . '\'');
  }
}
```

```
if (isset($_REQUEST['action']))
{
  if ('add' == $_REQUEST['action'])
  {
    $quantity = get_quantity($_REQUEST['item']);

    $quantity++;

    $q = $db->queryExec('UPDATE data SET quantity=' . $quantity . ' WHERE item=\'' .
         $_REQUEST['item'] . '\'');
  }

  if ('remove' == $_REQUEST['action'])
  {
    $quantity = get_quantity($_REQUEST['item']);

    if ($quantity > 0) $quantity--;

    $q = $db->queryExec('UPDATE data SET quantity=' . $quantity . ' WHERE item=\'' .
         $_REQUEST['item'] . '\'');
  }
}


function get_quantity($aItem)
{
  $db = new SQLiteDatabase('report.sqlite');

  $q = $db->query('SELECT quantity FROM data WHERE item=\'' . $aItem . '\'');
  $result = $q->current(SQLITE_NUM);

  return $result[0];
}
```

```
if (isset($_REQUEST['action']))
{
  if ('add' == $_REQUEST['action'])
  {
    $quantity = get_quantity($_REQUEST['item']);

    $quantity++;

    $q = $db->queryExec('UPDATE data SET quantity=' . $quantity . ' WHERE item=\'' .
         $_REQUEST['item'] . '\'');
  }

  if ('remove' == $_REQUEST['action'])
  {
    $quantity = get_quantity($_REQUEST['item']);

    if ($quantity > 0) $quantity--;

    $q = $db->queryExec('UPDATE data SET quantity=' . $quantity . ' WHERE item=\'' .
         $_REQUEST['item'] . '\'');
  }
}
```

```
if (isset($_REQUEST['action']))
{
  if ('add' == $_REQUEST['action'])
  {
    $quantity = get_quantity($_REQUEST['item']);

    $quantity++;

    update_quantity($_REQUEST['item'], $quantity);
  }

  if ('remove' == $_REQUEST['action'])
  {
    $quantity = get_quantity($_REQUEST['item']);

    if ($quantity > 0) $quantity--;

    update_quantity($_REQUEST['item'], $quantity);
  }
}


function update_quantity($aItem, $aQuantity)
{
  $db = new SQLiteDatabase('report.sqlite');

  $q = $db->queryExec('UPDATE data SET quantity=\'' . $aQuantity . '\' ' .
                      'WHERE item=\'' . $aItem . '\'');
}
```

```
if (isset($_REQUEST['action']))
{
  if ('add' == $_REQUEST['action'])
  {
    $quantity = get_quantity($_REQUEST['item']);

    $quantity++;

    update_quantity($_REQUEST['item'], $quantity);
  }

  if ('remove' == $_REQUEST['action'])
  {
    $quantity = get_quantity($_REQUEST['item']);

    if ($quantity > 0) $quantity--;

    update_quantity($_REQUEST['item'], $quantity);
  }
}
```

```
if (isset($_REQUEST['action']))
{
  if ('add' == $_REQUEST['action'])
  {
    $q = $db->query('SELECT quantity FROM
                     data WHERE item=\'' .
                     $_REQUEST['item'] . '\'');
    $quantity = $q->current(SQLITE_NUM);
    $quantity = $quantity[0];

    $quantity++;

    $q = $db->queryExec('UPDATE data SET
                         quantity=' . $quantity .
                         ' WHERE item=\'' .
         $_REQUEST['item'] . '\'');
  }
  if ('remove' == $_REQUEST['action'])
  {
    $q = $db->query('SELECT quantity FROM
                     data WHERE item=\'' .
                     $_REQUEST['item'] . '\'');
    $quantity = $q->current(SQLITE_NUM);
    $quantity = $quantity[0];

    if ($quantity > 0) $quantity--;

    $q = $db->queryExec('UPDATE data SET
                         quantity=' . $quantity .
                         ' WHERE item=\'' .
         $_REQUEST['item'] . '\'');
  }
}
```

```
<form name="contactform" method="POST">
<input type="hidden" name="contactform" value="true">
<table>

<tr>
  <td>Firstname:</td>
  <td><input type="text" name="firstname" value="<?php print $firstname; ?>"></td>
</tr>

<tr>
  <td>Lastname:</td>
  <td><input type="text" name="lastname" value="<?php print $lastname; ?>"></td>
</tr>

<tr>
  <td>Email:</td>
  <td><input type="text" name="email" value="<?php print $email; ?>"></td>
</tr>

...
```

```
<form name="contactform" method="POST">
<input type="hidden" name="contactform" value="true">
<table>

<tr>
  <td>Firstname:</td>
  <td><input type="text" name="firstname" value="<?php print $firstname; ?>"></td>
</tr>

<tr>
  <td>Lastname:</td>
  <td><input type="text" name="lastname" value="<?php print $lastname; ?>"></td>
</tr>

<tr>
  <td>Email:</td>
  <td><input type="text" name="email" value="<?php print $email; ?>"></td>
</tr>

...
```

```
<form name="contactform" method="POST">
<input type="hidden" name="contactform" value="true">
<table>

<tr>
  <td>Firstname:</td>
  <td><?php print get_input('firstname', $firstname); ?></td>
</tr>

<tr>
  <td>Lastname:</td>
  <td><?php print get_input('lastname', $lastname); ?></td>
</tr>

<tr>
  <td>Email:</td>
  <td><?php print get_input('email', $email); ?></td>
</tr>

...


function get_input($aName, $aValue)
{
  return '<input type="text" name="' . $aName . '" value="' . $aValue . '">';
}
```

```
<form name="contactform" method="POST">
<input type="hidden" name="contactform" value="true">
<table>

<tr>
  <td>Firstname:</td>
  <td><?php print get_input('firstname', $firstname); ?></td>
</tr>

<tr>
  <td>Lastname:</td>
  <td><?php print get_input('lastname', $lastname); ?></td>
</tr>

<tr>
  <td>Email:</td>
  <td><?php print get_input('email', $email); ?></td>
</tr>

...
```

```
<form name="contactform" method="POST">
<input type="hidden" name="contactform" value="true">
<table>

<?php print get_form_row('Firstname:', get_input('firstname', $firstname)); ?>

<?php print get_form_row('Lastname:', get_input('lastname', $lastname)); ?>

<?php print get_form_row('Email:', get_input('email', $email)); ?>

...


function get_form_row($aLabel, $aField)
{
  return '<tr><td>' . $aLabel . '</td><td>' . $aField . '</td></tr>';
}
```

```
<form name="contactform" method="POST">
<input type="hidden" name="contactform" value="true">
<table>

<?php print get_form_row('Firstname:', get_input('firstname', $firstname)); ?>

<?php print get_form_row('Lastname:', get_input('lastname', $lastname)); ?>

<?php print get_form_row('Email:', get_input('email', $email)); ?>

...
```

Shorten Code Blocks

- Break down code into little pieces
- A method or function should fit on the screen without scrolling
- When the code needs (too many) inline comments, it may be too complex
- No more than three nesting levels

The 30 Second Rule

If you don't understand a function or method after looking at the code for 30 seconds, you need to refactor

Separate Different Concerns

- Three-Tier architecture:
  - Presentation
  - Business Logic
  - Data Access
- Functionality (add, remove, modify)
- Input (pre)processing
- No SQL near HTML

Separate Different Concerns

- Makes Unit Tests possible, since the logic is decoupled from data source and presentation
- Much more code reuse
- Code becomes easier to read because you can focus on one aspect

```
$q = $db->query('SELECT * from data ORDER BY ' . $order);
print '<table>';

print '<tr><td><b><a href="?sort=item">Item</a></b></td><td><b><a
href="?sort=barcode">Barcode</a></b></td><td><b><a
href="?sort=shelf">Shelf</a></b></td><td><b><a
href="?sort=quantity">Quantity</a></b></td></tr>';

while($q->valid())
{
  $r = $q->current(SQLITE_ASSOC);

  print '<tr>';
  $first = true;

  foreach ($r as $key => $value)
  {
    if ($first)
    {
      $item = $value;
      $first = false;
    }

    print '<td>' . $value . '</td>';

    if ('quantity' == $key) $sum += $value;
  }

  print '<td>[<a href="?sort=' . $order . '&action=add&item=' . $item . '">+</a>]</td><td>[<a
href="?sort=' . $order . '&action=remove&item=' . $item . '">-</a>]</td>';

  print '</tr>';
  $q->next();
}

print '<tr><td></td><td></td><td>Total:</td><td><b>' . $sum . '</b></td></tr></table>';
```

```
$q = $db->query('SELECT * from data ORDER BY ' . $order);
print '<table>';

print '<tr><td><b><a href="?sort=item">Item</a></b></td><td><b><a
href="?sort=barcode">Barcode</a></b></td><td><b><a
href="?sort=shelf">Shelf</a></b></td><td><b><a
href="?sort=quantity">Quantity</a></b></td></tr>';

while($q->valid())
{
  $r = $q->current(SQLITE_ASSOC);

  print '<tr>';
  $first = true;

  foreach ($r as $key => $value)
  {
    if ($first)
    {
      $item = $value;
      $first = false;
    }

    print '<td>' . $value . '</td>';

    if ('quantity' == $key) $sum += $value;
  }

  print '<td>[<a href="?sort=' . $order . '&action=add&item=' . $item . '">+</a>]</td><td>[<a
href="?sort=' . $order . '&action=remove&item=' . $item . '">-</a>]</td>';

  print '</tr>';
  $q->next();
}

print '<tr><td></td><td></td><td>Total:</td><td><b>' . $sum . '</b></td></tr></table>';
```

```
$q = $db->query('SELECT * from data ORDER BY ' . $order);
print '<table>';

print '<tr><td><b><a href="?sort=item">Item</a></b></td><td><b><a
href="?sort=barcode">Barcode</a></b></td><td><b><a
href="?sort=shelf">Shelf</a></b></td><td><b><a
href="?sort=quantity">Quantity</a></b></td></tr>';

while($q->valid())
{
  $r = $q->current(SQLITE_ASSOC);

  print '<tr>';
  $first = true;

  foreach ($r as $key => $value)
  {
    if ($first)
    {
      $item = $value;
      $first = false;
    }

    print '<td>' . $value . '</td>';

    if ('quantity' == $key) $sum += $value;
  }

  print '<td>[<a href="?sort=' . $order . '&action=add&item=' . $item . '">+</a>]</td><td>[<a
href="?sort=' . $order . '&action=remove&item=' . $item . '">-</a>]</td>';

  print '</tr>';
  $q->next();
}

print '<tr><td></td><td></td><td>Total:</td><td><b>' . $sum . '</b></td></tr></table>';
```

```
$q = $db->query('SELECT * from data ORDER BY ' . $order);
print '<table>';

print '<tr><td><b><a href="?sort=item">Item</a></b></td><td><b><a
href="?sort=barcode">Barcode</a></b></td><td><b><a
href="?sort=shelf">Shelf</a></b></td><td><b><a
href="?sort=quantity">Quantity</a></b></td></tr>';

while($q->valid())
{
  $r = $q->current(SQLITE_ASSOC);

  print '<tr>';
  $first = true;

  foreach ($r as $key => $value)
  {
    if ($first)
    {
      $item = $value;
      $first = false;
    }

    print '<td>' . $value . '</td>';

    if ('quantity' == $key) $sum += $value;
  }

  print '<td>[<a href="?sort=' . $order . '&action=add&item=' . $item . '">+</a>]</td><td>[<a
href="?sort=' . $order . '&action=remove&item=' . $item . '">-</a>]</td>';

  print '</tr>';
  $q->next();
}

print '<tr><td></td><td></td><td>Total:</td><td><b>' . $sum . '</b></td></tr></table>';
```

```
function get_data($aSort)
{
  $db = new SQLiteDatabase('report.sqlite');

  $q = $db->query('SELECT * from data ORDER BY ' . $aSort);

  $result = array();

  while($q->valid())
  {
    $result[] = $q->current(SQLITE_ASSOC);
    $q->next();
  }


  return $result;
}
```

```
print '<table>';

print '<tr><td><b><a href="?sort=item">Item</a></b></td><td><b><a
href="?sort=barcode">Barcode</a></b></td><td><b><a
href="?sort=shelf">Shelf</a></b></td><td><b><a
href="?sort=quantity">Quantity</a></b></td></tr>';

foreach (get_data() as $r)
{
  print '<tr>';
  $first = true;

  foreach ($r as $key => $value)
  {
    if ($first)
    {
      $item = $value;
      $first = false;
    }

    print '<td>' . $value . '</td>';

    if ('quantity' == $key) $sum += $value;
  }

  print '<td>[<a href="?sort=' . $order . '&action=add&item=' . $item . '">+</a>]</td><td>[<a
href="?sort=' . $order . '&action=remove&item=' . $item . '">-</a>]</td>';

  print '</tr>';
}

print '<tr><td></td><td></td><td>Total:</td><td><b>' . $sum . '</b></td></tr></table>';
```

```
print '<table>';

print '<tr><td><b><a href="?sort=item">Item</a></b></td><td><b><a
href="?sort=barcode">Barcode</a></b></td><td><b><a
href="?sort=shelf">Shelf</a></b></td><td><b><a
href="?sort=quantity">Quantity</a></b></td></tr>';

foreach (get_data() as $r)
{
  print '<tr>';
  $first = true;

  foreach ($r as $key => $value)
  {
    if ($first)
    {
      $item = $value;
      $first = false;
    }

    print '<td>' . $value . '</td>';

    if ('quantity' == $key) $sum += $value;
  }

  print '<td>[<a href="?sort=' . $order . '&action=add&item=' . $item . '">+</a>]</td><td>[<a
href="?sort=' . $order . '&action=remove&item=' . $item . '">-</a>]</td>';

  print '</tr>';
}

print '<tr><td></td><td></td><td>Total:</td><td><b>' . $sum . '</b></td></tr></table>';
```

```
function get_total()
{
  $db = new SQLiteDatabase('report.sqlite');

  $q = $db->query('SELECT SUM(quantity) from data');
  $result = $q->current(SQLITE_NUM);

  return $result[0];
}
```

```
print '<table>';

print '<tr><td><b><a href="?sort=item">Item</a></b></td><td><b><a
href="?sort=barcode">Barcode</a></b></td><td><b><a
href="?sort=shelf">Shelf</a></b></td><td><b><a
href="?sort=quantity">Quantity</a></b></td></tr>';

foreach (get_data() as $r)
{
  print '<tr>';
  $first = true;

  foreach ($r as $key => $value)
  {
    if ($first)
    {
      $item = $value;
      $first = false;
    }

    print '<td>' . $value . '</td>';
  }

  print '<td>[<a href="?sort=' . $order . '&action=add&item=' . $item . '">+</a>]</td><td>[<a
href="?sort=' . $order . '&action=remove&item=' . $item . '">-</a>]</td>';

  print '</tr>';
}

print '<tr><td></td><td></td><td>Total:</td><td><b>' . get_total() . '</b></td></tr></table>';
```

```
print create_good_old_HTML_table(get_data(), get_total());

print create_fancy_AJAX_user_interface(get_data(), get_total());

export_data_to_a_really_really_special_format(get_data());
```

```
<form name="contactform" method="POST">
<input type="hidden" name="contactform" value="true">
<table>

<?php print get_form_row('Firstname:', get_input('firstname', $firstname)); ?>

<?php print get_form_row('Lastname:', get_input('lastname', $lastname)); ?>

<?php print get_form_row('Email:', get_input('email', $email)); ?>

...
```

```
function create_form_body($aFormRows)
{
  $result = '';

  foreach($aFormRows as $row)
  {
    $result .= get_form_row($row[0], $row[1]);
  }

  return $result;
}
```

```
<form name="contactform" method="POST">
<input type="hidden" name="contactform" value="true">
<table>

<?php

  $form_rows = array(
    array('Firstname:', get_input('firstname', $firstname)),
    array('Lastname:', get_input('lastname', $lastname)),
    array('Email:', get_input('email', $email)),
  );


  print create_form_body($form_rows)

?>

...
```

```
<form name="contactform" method="POST">
<input type="hidden" name="contactform" value="true">
<table>

<?php

  $form_rows = array(
    array('Firstname:', get_input('firstname', $firstname)),
    array('Lastname:', get_input('lastname', $lastname)),
    array('Email:', get_input('email', $email)),
  );


  print create_form($form_rows)

?>

...
```

```
$form_rows = array(
  array('Firstname:', get_input('firstname', $firstname)),
  array('Lastname:', get_input('lastname', $lastname)),
  array('Email:', get_input('email', $email)),
);


print create_form('contactform', $form_rows);


function create_form($aName, $aFormBody)
{
  $result = '<form name="contactform" method="POST">' .
            '<input type="hidden" name="contactform" value="true">' .
            '<table>';

  $result .= create_form_body($aFormBody);

  $result .= ...;

  return $result;
}
```

```
$form_rows = array(
  array('Firstname:', get_input('firstname', $firstname)),
  array('Lastname:', get_input('lastname', $lastname)),
  array('Email:', get_input('email', $email)),
);

print create_form('contactform', $form_rows);
```

```
if (isset($_REQUEST['action']))
{
  if ('add' == $_REQUEST['action'])
  {
    $quantity = get_quantity($_REQUEST['item']);

    $quantity++;

    update_quantity($_REQUEST['item'], $quantity);
  }

  if ('remove' == $_REQUEST['action'])
  {
    $quantity = get_quantity($_REQUEST['item']);

    if ($quantity > 0) $quantity--;

    update_quantity($_REQUEST['item'], $quantity);
  }
}
```

```
$item = isset($_REQUEST['item'])? $_REQUEST['item'] : '';


if (isset($_REQUEST['action']))
{
  if ('add' == $_REQUEST['action'])
  {
    $quantity = get_quantity($item);

    $quantity++;

    update_quantity($item], $quantity);
  }

  if ('remove' == $_REQUEST['action'])
  {
    $quantity = get_quantity($item);

    if ($quantity > 0) $quantity--;

    update_quantity($item, $quantity);
  }
}
```

```php
$item = isset($_REQUEST['item'])? $_REQUEST['item'] : '';


if (isset($_REQUEST['action']))
{
  if ('add' == $_REQUEST['action'])
  {
    $quantity = get_quantity($item);

    $quantity++;

    update_quantity($item], $quantity);
  }

  if ('remove' == $_REQUEST['action'])
  {
    $quantity = get_quantity($item);

    if ($quantity > 0) $quantity--;

    update_quantity($item, $quantity);
  }
}
```

```
$item = isset($_REQUEST['item'])? $_REQUEST['item'] : '';
$action = isset($_REQUEST['action']) ? $_REQUEST['action'] : '';


if ('' != $action))
{
  if ('add' == $action)
  {
    $quantity = get_quantity($item);

    $quantity++;

    update_quantity($item], $quantity);
  }

  if ('remove' == $action)
  {
    $quantity = get_quantity($item);

    if ($quantity > 0) $quantity--;

    update_quantity($item, $quantity);
  }
}
```

```
$item = isset($_REQUEST['item'])? $_REQUEST['item'] : '';
$action = isset($_REQUEST['action']) ? $_REQUEST['action'] : '';


if ('' != $action))
{
  if ('add' == $action)
  {
    $quantity = get_quantity($item);

    $quantity++;

    update_quantity($item], $quantity);
  }

  if ('remove' == $action)
  {
    $quantity = get_quantity($item);

    if ($quantity > 0) $quantity--;

    update_quantity($item, $quantity);
  }
}
```

```
$item = isset($_REQUEST['item'])? $_REQUEST['item'] : '';
$action = isset($_REQUEST['action']) ? $_REQUEST['action'] : '';


if ('add' == $action)
{
  $quantity = get_quantity($item);

  $quantity++;

  update_quantity($item], $quantity);
}

if ('remove' == $action)
{
  $quantity = get_quantity($item);

  if ($quantity > 0) $quantity--;

  update_quantity($item, $quantity);
}
```

```
$item = isset($_REQUEST['item'])? $_REQUEST['item'] : '';
$action = isset($_REQUEST['action']) ? $_REQUEST['action'] : '';


if ('add' == $action)
{
  $quantity = get_quantity($item);

  $quantity++;

  update_quantity($item], $quantity);
}

if ('remove' == $action)
{
  $quantity = get_quantity($item);

  if ($quantity > 0) $quantity--;

  update_quantity($item, $quantity);
}
```

```
$item = get_input('item');
$action = get_input('action');


if ('add' == $action)
{
  $quantity = get_quantity($item);

  $quantity++;

  update_quantity($item], $quantity);
}

if ('remove' == $action)
{
  $quantity = get_quantity($item);

  if ($quantity > 0) $quantity--;

  update_quantity($item, $quantity);
}


function get_input($aParameter)
{
  if (!isset($_REQUEST[$aParameter])) return '';

  return $_REQUEST[$aParameter];
}
```

```
$item = get_input('item');
$action = get_input('action');


if ('add' == $action)
{
  $quantity = get_quantity($item);

  $quantity++;

  update_quantity($item], $quantity);
}

if ('remove' == $action)
{
  $quantity = get_quantity($item);

  if ($quantity > 0) $quantity--;

  update_quantity($item, $quantity);
}
```

```
if (isset($_REQUEST['action']))
{
  if ('add' == $_REQUEST['action'])
  {
    $quantity = get_quantity($_REQUEST['item']);

    $quantity++;

    update_quantity($_REQUEST['item'], $quantity);
  }

  if ('remove' == $_REQUEST['action'])
  {
    $quantity = get_quantity($_REQUEST['item']);

    if ($quantity > 0) $quantity--;

    update_quantity($_REQUEST['item'], $quantity);
  }
}
```

Replace Implementations

- Use native PHP instead of an old extension
- Use components
  (PEAR, Zend Framework, ezComponents)
- Use PHP extensions
  (XMLWriter, SOAP)

Rewrite Ugly Code

- Rewrite ugly functions/methods
- If you don't rely on global variables, this should not affect the rest of the application

The Seven Steps

1. Format Source Code consistently
2. Use Consistent Naming
3. Document the API
4. Eliminate Redundant Code
5. Shorten Code Blocks
6. Separate Different Concerns
7. Replace Implementations

Thank you. Questions?

- Presentation Slides will be made available at http://www.zend.com/resources/webinars
- Audio and slides of the first part are also available
- Email me at stefan.priebsch@e-novative.de
- Have a Nice Day ...
  ... and try some Refactoring today!