OpenERP
OPEN SOURCE BUSINESS APPLICATIONS

OpenERP v6.0
Technical Improvements

Olivier Dony

OpenERP
OPEN SOURCE BUSINESS APPLICATIONS

Agenda

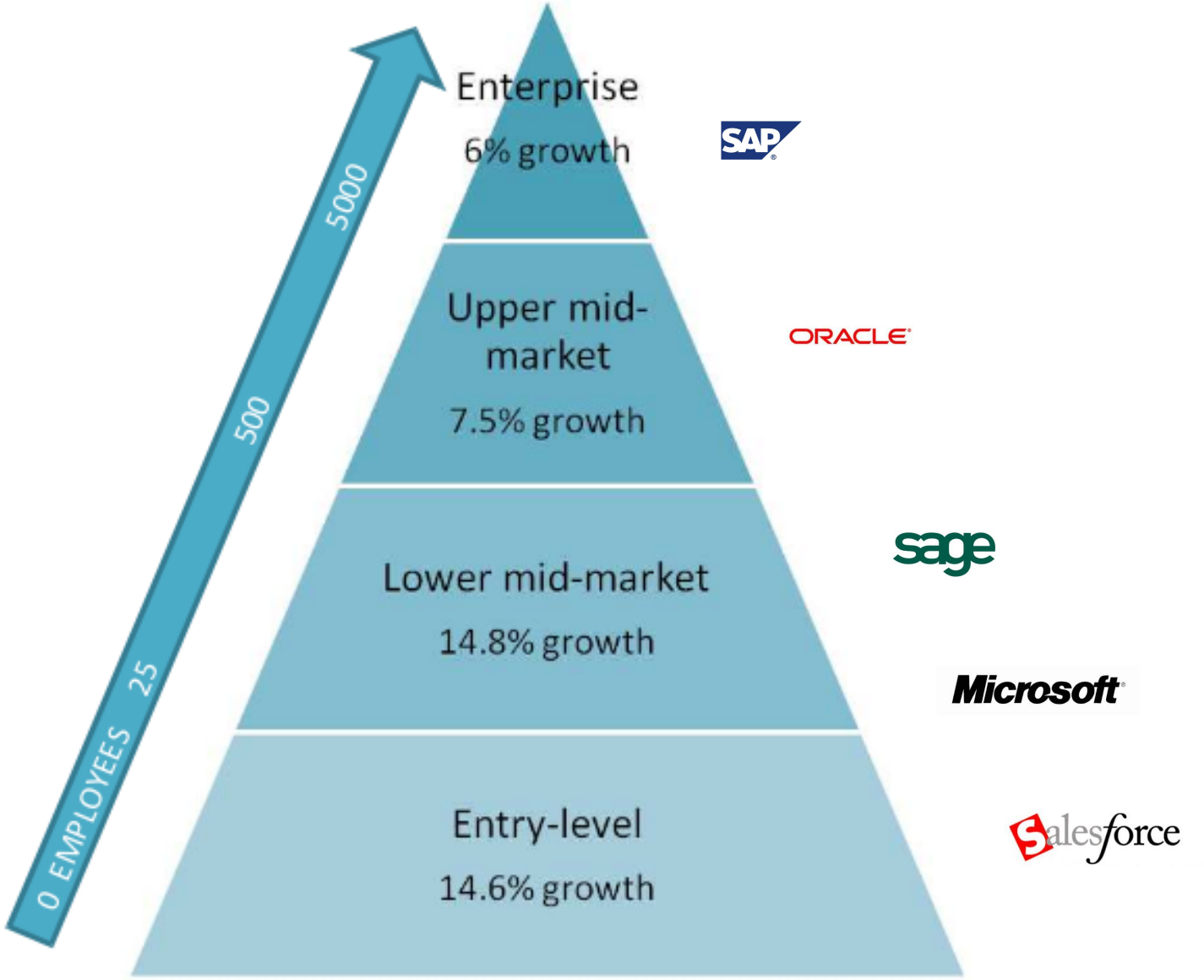
- Part 1
  - Short overview
  - Search views
  - Improved Tree views, aggregation, dynamic reports
  - Diagram views
  - Multi-company
- Part 2
  - Testing framework and continuous integration
  - Usability guidelines
  - New R&D Process
  - Web add-ons
  - Miscellaneous improvements and tips

OpenERP
OPEN SOURCE BUSINESS APPLICATIONS

Agenda

- Part 1
  - Short overview
  - Search views
  - Improved Tree views, aggregation, dynamic reports
  - Diagram views
  - Multi-company
- Part 2
  - Testing framework and continuous integration
  - Usability guidelines
  - New R&D Process
  - Web add-ons
  - Miscellaneous improvements and tips

OpenERP is disruptive



OpenERP is disruptive

Disruptive product

- Modern programming language
- Modular approach
- Fully web-based
- Classical or cloud deployment

Disruptive business

- Free download
- Contributions and Community fuel the product
- Value created is fairly split between Customer, Service Provider, and Publisher

2010 = Cornerstone for OpenERP

- During the last 5 years, OpenERP has matured into a **smart Enterprise Management software suite**, with a **world-wide** brand, and a **strong market** demand.
- OpenERP SA now puts focus on being the OpenERP Publisher
- OpenERP v6 focuses on **ease-of-use** for these smaller companies!

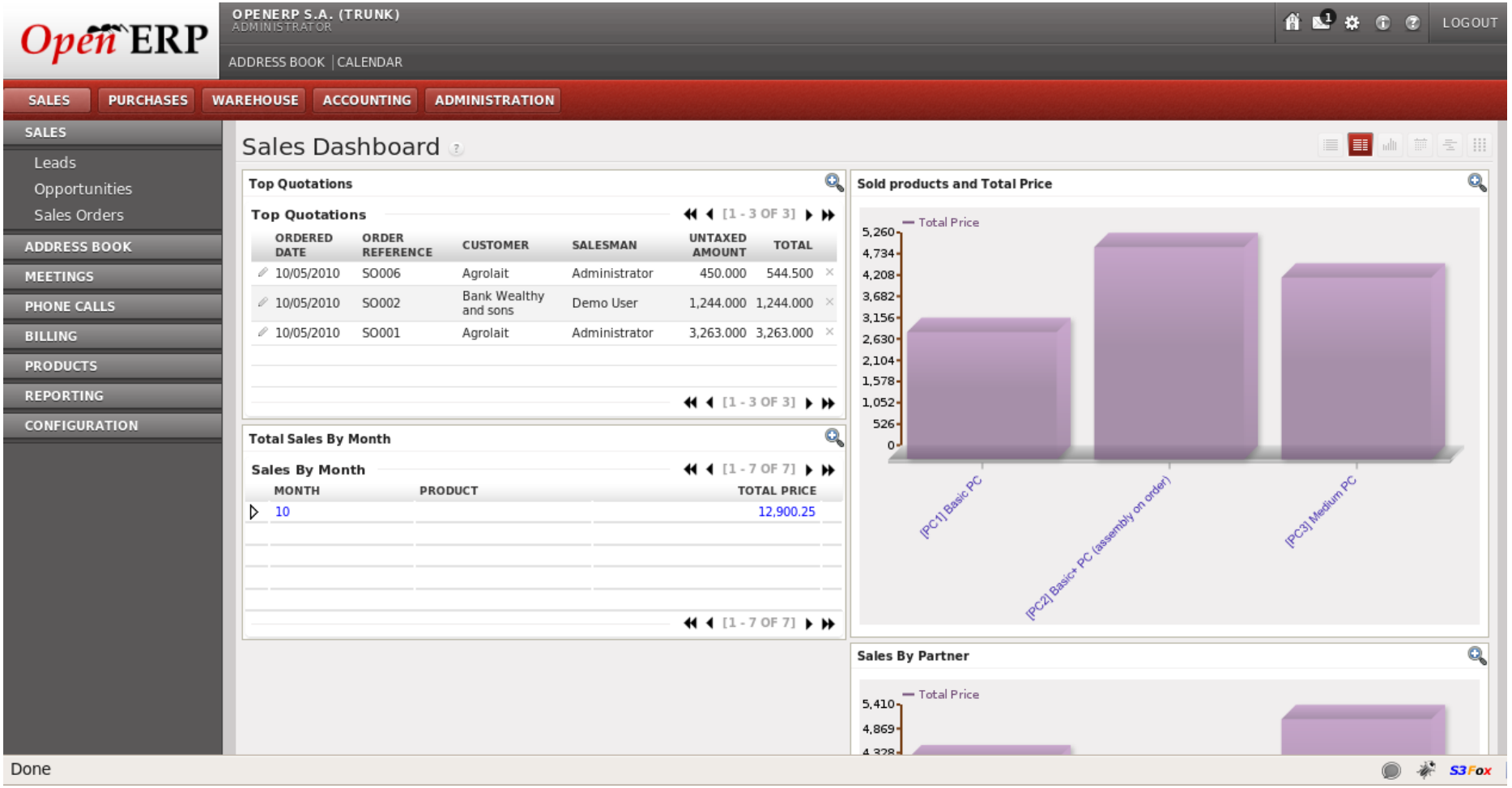
Technical Goals for OpenERP v6.0

- Make OpenERP **easier** for smaller companies
- Global **code review** and simplification
- New **features**
- **New** development **processes**

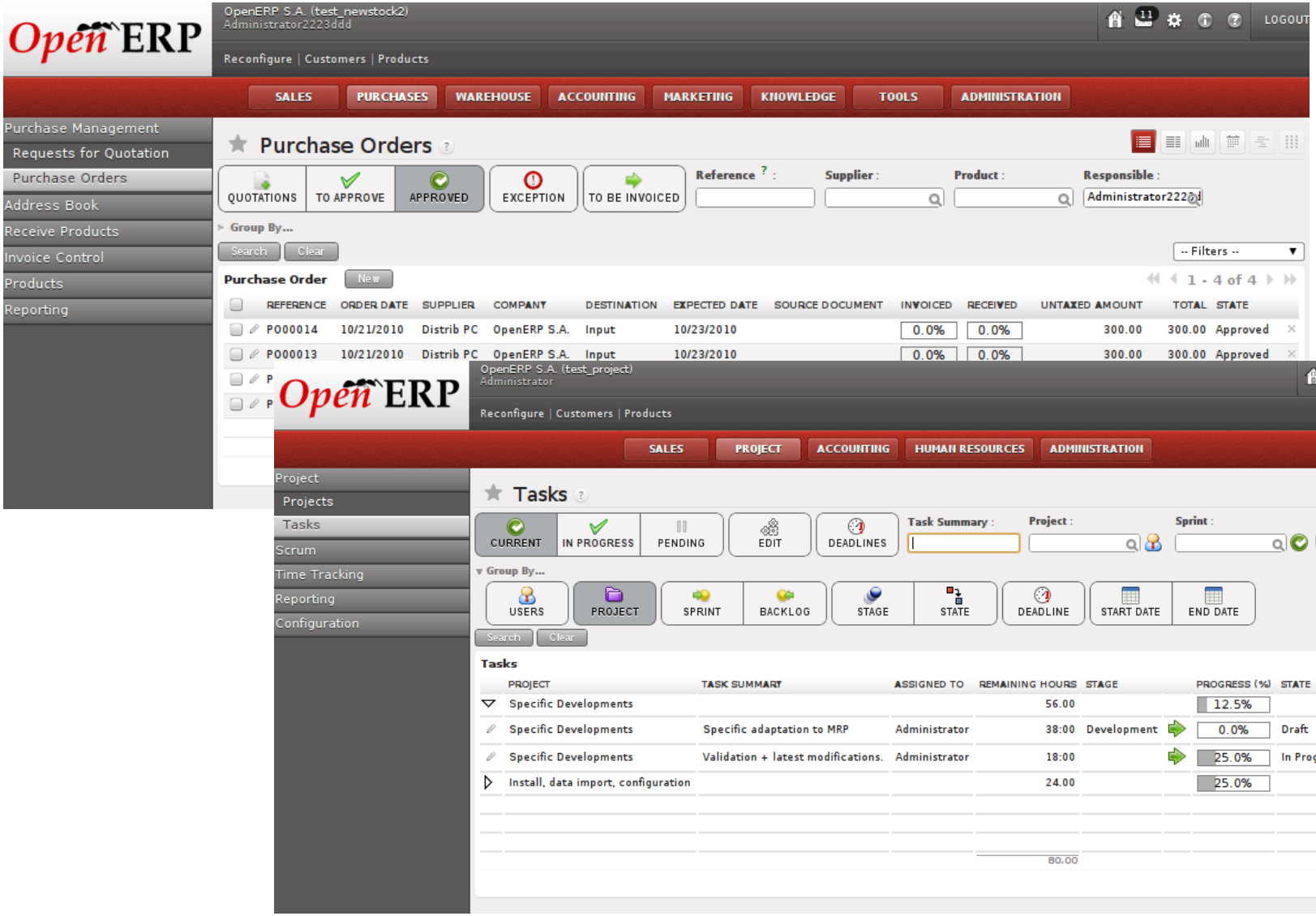
Easier for smaller companies

- OpenERP becomes a true **suite of Business applications**
  - Menus and modules structure reviewed and split
  - Smaller, specific applications
    - » account = invoicing only
    - » account_accountant = full financial management
- Simplified views (vs. extended)
- Business-oriented screens and filters

Business Applications



Business-oriented screens



Code Review & Simplifications

- Improved modularity
  - e.g: procurement (~~follows~~), analytic, resource
- Global Code Review
  - Modules rewritten: crm, hr, caldav,
  - Performance improvements: lists, reports
  - Lint cleanup
  - Simpler aggregation/reporting objects
- Hundreds of tests
- Lots of source code lines removed

New R&D Processes

- Dedicated R&D Teams
- SCRUM methodology
- Bug qualification team
- Testing tools and test suite
- Continuous integration system: Buildbot

More on this in Part 2...

New features

- New views: search, diagram, improved lists
- Improved API – core objects
- Focus on reusable and generic features
  - custom filters
  - mail.gateway, caldav, resources
- Multi-company built-in

» Let's zoom in...

OpenERP
OPEN SOURCE BUSINESS APPLICATIONS

Agenda

- Part 1
  - Short overview
  - Search views
  - Improved Tree views, aggregation, dynamic reports
  - Diagram views
  - Multi-company
- Part 2
  - Testing framework and continuous integration
  - Usability guidelines
  - New R&D Process
  - Web add-ons
  - Miscellaneous improvements and tips

Search Views

- Improves over built-in search features in v5
- Always associated with other view: list, calendar, …
- Deprecates ~~`select=1`~~, ~~`select=2`~~ → `select = True`
- New view type: `<search>`
- Two main building blocks:
  - `<field name='...'>`
  - `<filter domain='...' context='...'>`
- Structured with:
  - `<group>`
  - `<separator>`
  - `<newline>`

Search Views

```
<search string="Search Project">
   <group col='15' colspan='4'>
       <filter icon="terp-check" string="Current" name="current"
               domain="[('state', 'in',('open','pending'))]"
               help="Open and Pending Projects"/>
       <separator orientation="vertical"/>
       <field name="name" string="Project Name"/>
       <field name="user_id" string="Project Manager">
           <filter domain="[('user_id','=',uid)]" icon="terp-personal"
                   help="Projects in which I am a manage" />
       </field>
       <field name="partner_id" string="Partner"/>
   </group>
   <newline />
   <group expand="0" string="Group By..." colspan="4" col="20"
          groups="base.group_extended">
        <filter string="Manager" name="Manager" icon="terp-personal"
                domain = "[]" context="{'group_by':'user_id'}"/>
        <separator orientation="vertical"/>
        <filter string="Parent" name="Parent" help="Parent Project"
                icon="terp-folder-blue" domain = "[]"
                context="{'group_by':'parent_id'}"/>
   </group>
</search>
```

Advanced Search Views

- <fields> with embedded <filters>
- Advanced attributes for <field>:
  - context="{'country_visible': *True*}"
  - domain="[('country_id', '=', 'Belgium')]"
  - filter_domain="['|', ('location_to', '=', **self**), ('location_src', '=', **self**)]"
- Default values for filters and fields, via action context:
  - context="{'search_default_filter1': True}"
  - context="{'search_default_field1': 'test'}"
  - context="{'search_default_field1': lang}"
- Specific search view can be forced in caller action:
  - <field name="search_view_id" ref="view_project_project_filter"/>

Search views: toggle group-by

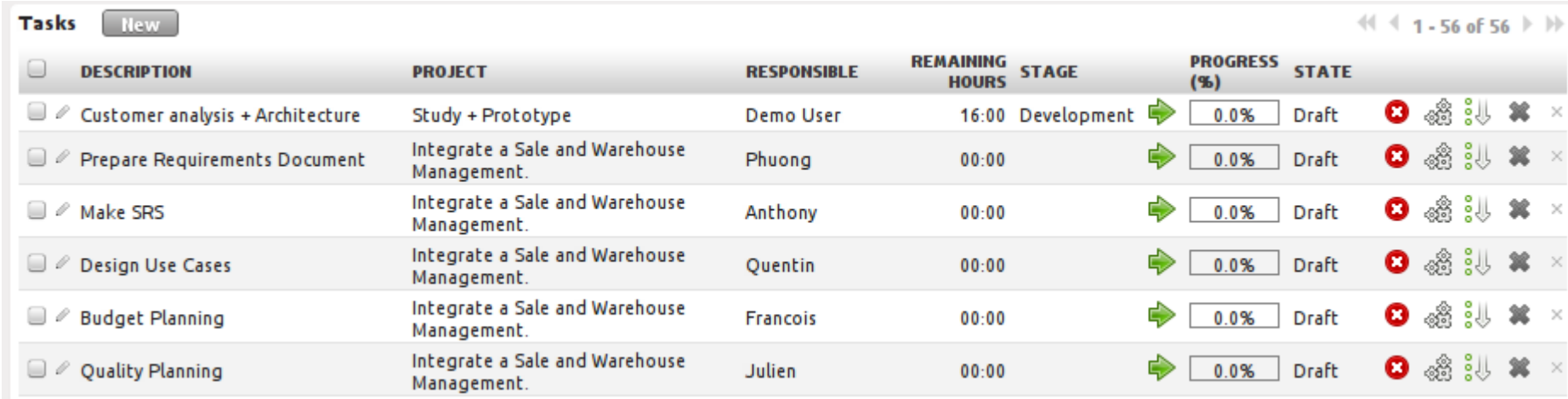
- `context` attribute of elements can toggle aggregation of associated list views
  - ```
    <filter string="Manager"
            name="manager"
            icon="terp-personal"
            context="{'group_by':'user_id'}"/>
    ```
- Multiple group-by levels are supported
  - ```
    <filter string="Manager,State"
            name="manager_state"
            context="{'group_by':['user_id','state']}"/>
    ```
- Order of group-by entries matters!

OpenERP
OPEN SOURCE BUSINESS APPLICATIONS

Agenda

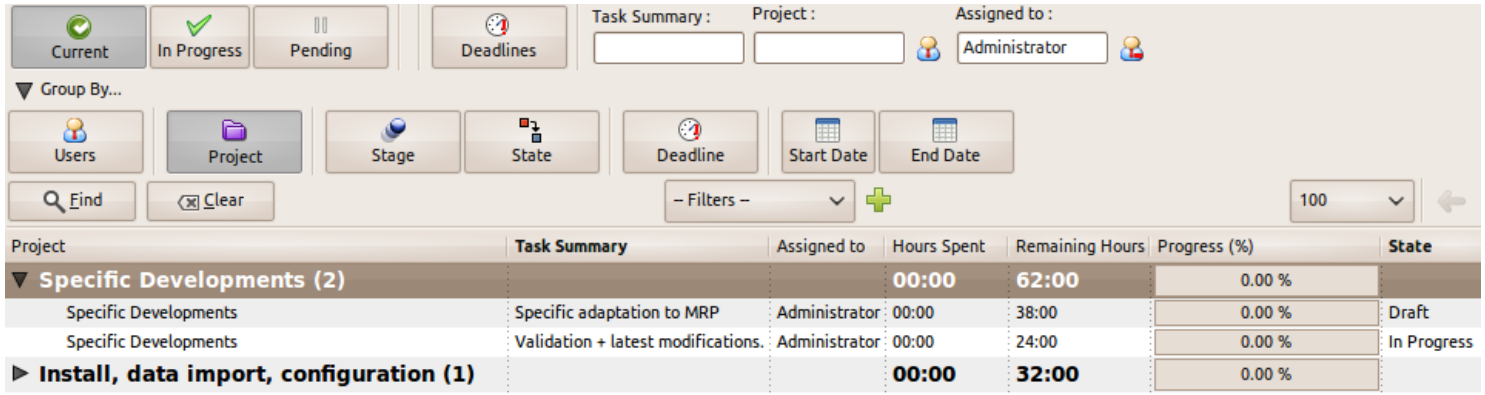
- Part 1
  - Short overview
  - Search views
  - Improved Tree views, aggregation, dynamic reports
  - Diagram views
  - Multi-company
- Part 2
  - Testing framework and continuous integration
  - Usability guidelines
  - New R&D Process
  - Web add-ons
  - Miscellaneous improvements and tips

Improved Lists



- List views can now display buttons
  - Regular `<button>` elements as in form views
  - Action/Method/Workflow called on single record
- Columns can be toggled via context
  - `context = {'section_mode': True}`
  - `<field name="country_id" invisible="context.get('invisible_country')">`

Improved Lists



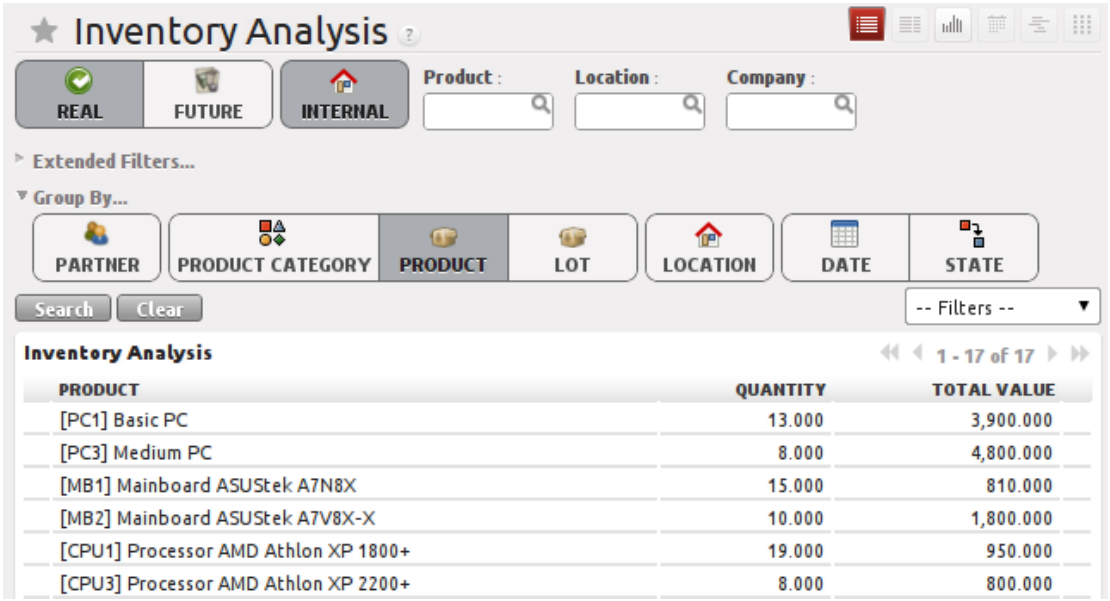
- List views now support aggregation aka group-by
  - Simple or multiple level aggregation
  - Controlled via context: e.g. search view or caller action

    ```
    context = {'group_by':['field_1','field_2']}
    ```

  - All contexts are combined
  - Aggregated columns shown on group header
    - » Numeric fields summed by default
    - » Set `group_operator` in `_columns` to customize (avg, min, …)
- New server-side API method: `read_group()`

Group-by = Real-time Reporting!

- Combine `group_by` values to get quick and intuitive real-time analysis
- When individual records are too low-level or irrelevant, hide them (e.g. stock moves) :
  - `context = {'group_by_no_leaf': 1}`



Building Reporting Objects

```
34 class crm_lead_report(osv.osv):
35     """ CRM Lead Report """
36     _name = "crm.lead.report"
37     _auto = False
38     _description = "CRM Lead Report"
39 
40     _columns = {
41         'name': fields.char('Year', size=64, required=False, readonly=True),
42         'user_id':fields.many2one('res.users', 'User', readonly=True),
43         'country_id':fields.many2one('res.country', 'Country', readonly=True),
44         'section_id':fields.many2one('crm.case.section', 'Sales Team', readonly=True),
45         'channel_id':fields.many2one('res.partner.canal', 'Channel', readonly=True),
 ...
80     def init(self, cr):
81         tools.drop_view_if_exists(cr, 'crm_lead_report')
82         cr.execute("""
83             CREATE OR REPLACE VIEW crm_lead_report AS (
84                 SELECT
85                     id,
86                     to_char(c.create_date, 'YYYY') as name,
87                     to_char(c.create_date, 'MM') as month,
88                     to_char(c.create_date, 'YYYY-MM-DD') as day,
89                     to_char(c.create_date, 'YYYY-MM-DD') as creation_date,
90                     to_char(c.date_open, 'YYYY-MM-DD') as opening_date,
91                     to_char(c.date_closed, 'YYYY-mm-dd') as date_closed,
92                     c.state,
 ...
113                 FROM
114                     crm_lead c
115             )""")
116 
117 crm_lead_report()
```

OpenERP
OPEN SOURCE BUSINESS APPLICATIONS

Agenda

- Part 1
  - Short overview
  - Search views
  - Improved Tree views, aggregation, dynamic reports
  - Diagram views
  - Multi-company
- Part 2
  - Testing framework and continuous integration
  - Usability guidelines
  - New R&D Process
  - Web add-ons
  - Miscellaneous improvements and tips

Diagram Views

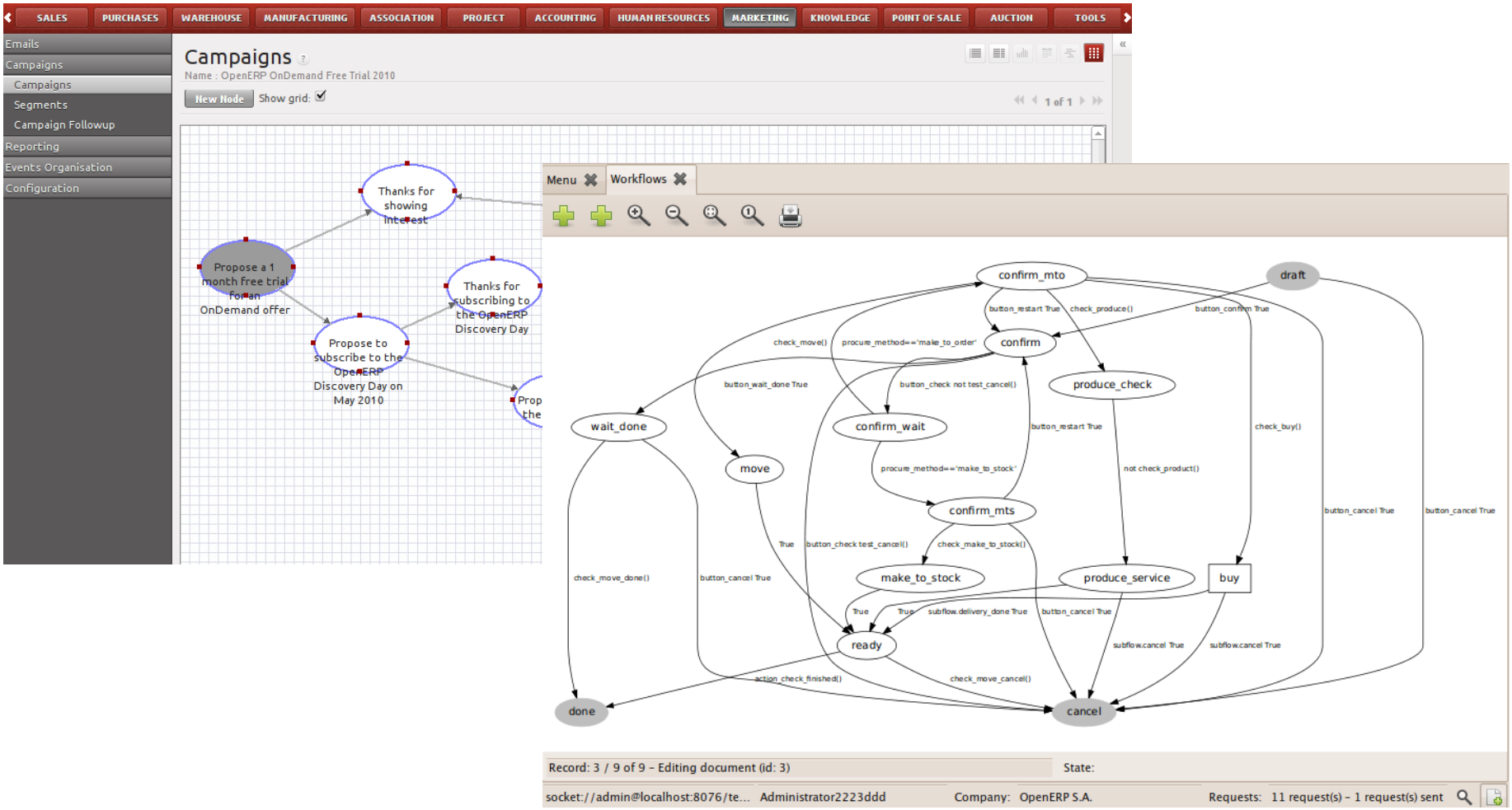
- Workflow Editor existed in Web Client 5.0
- Transformed into generic view, available in GTK too
- New <diagram> view type
- Constructed with
  - <node object="node.model"
    shape="..."
    bgcolor="..."><
  - <arrow object="transition.model"
    source="source_field"
    destination="dest_field" >
  - <field> (included in quick view)
- Objects must form a coherent graph
- <node> model must have a many2one to its container

Diagram Views

```
<diagram string="Workflow Editor">
    <node object="workflow.activity" shape="rectangle:subflow_id!=False"
          bgcolor="gray:flow_start==True;grey:flow_stop==True">
        <field name="name"/>
        <field name="kind"/>
        <field name="action"/>
        <field name="flow_start" invisible="1"/>
        <field name="flow_stop" invisible="1"/>
        <field name="subflow_id" invisible="1"/>
    </node>
    <arrow object="workflow.transition" source="act_from" destination="act_to"
           label="['signal','condition']">
        <field name="act_from"/>
        <field name="act_to"/>
        <field name="signal"/>
    </arrow>
</diagram>
```

Diagram Views

- Allows graphical definition of complex structures



OpenERP
OPEN SOURCE BUSINESS APPLICATIONS

Agenda

- Part 1
  - Short overview
  - Search views
  - Improved Tree views, aggregation, dynamic reports
  - Diagram views
  - Multi-company
- Part 2
  - Testing framework and continuous integration
  - Usability guidelines
  - New R&D Process
  - Web add-ons
  - Miscellaneous improvements and tips

Multi-Company

- Now built-in in all applications
- multi_company module = demo data only
- in-depth review added specific internal operations:
  - e.g. for warehouse management:
    - » inventory push/pull flows, transit locations
    - » multi-company invoice control based on push/pull flows
    - » More on this in functional training (tomorrow PM)

Multi-company

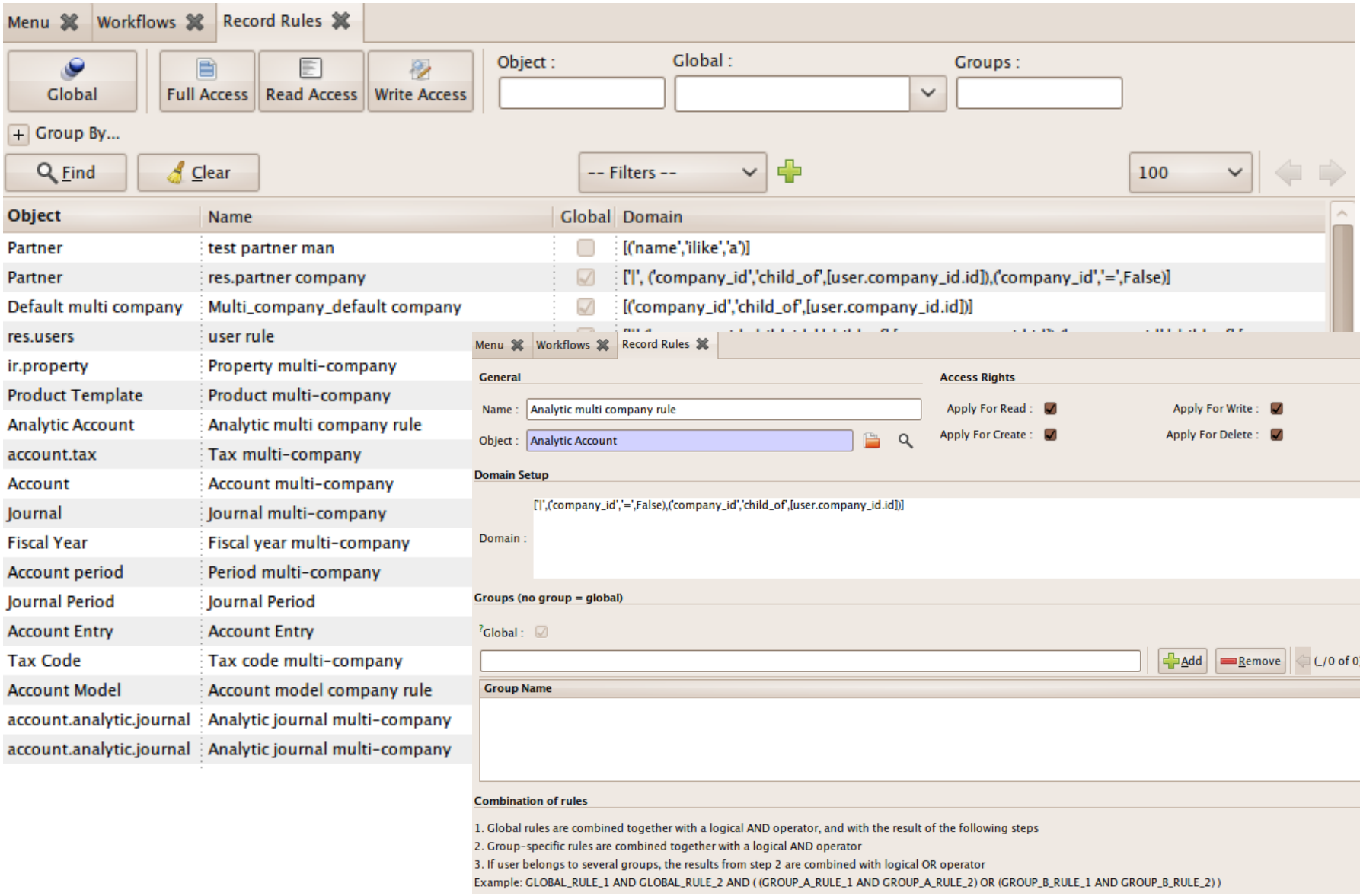
- Many models now have a direct or indirect link to a company (many2one or related field)
  - e.g: sale.order.line →sale.order →shop →company
- Relevant company is assigned when creating new records
- Special group is needed to view multi-company fields
  - Useability/Multi Companies
- Users can belong to several companies, but select the one the current context they work in, via Preferences
- Visibility of records depends on Company they belong to, current Company of user, and Record Rules.

Multi-company

- Record rules have been improved since v5.0
- No more rule groups – directly linked to user groups
- Now controlled per mode: create/read/update/delete
- Strong security semantics between “global” and “local” (group-specific) rules
  - Global rules cannot be bypassed under any circumstance, and always restrict the access
  - Local rules are additive and grant more access

```
global1 AND global2 AND (local1 OR local2)
```

Multi-company: Security



OpenERP
OPEN SOURCE BUSINESS APPLICATIONS

Agenda

- Part 1
  - Short overview
  - Search views
  - Improved Tree views, aggregation, dynamic reports
  - Diagram views
  - Multi-company
- Part 2
  - Testing framework and continuous integration
  - Usability guidelines
  - New R&D Process
  - Web add-ons
  - Miscellaneous improvements and tips

Testing framework: YAML

- New YAML data serialization supported in v6
- Data-oriented syntax *"YAML Ain't a Markup Language"*
- Whitespace delimiters, Python-like
- Human-readable
  - no quotes, brackets, braces, … by default
- Works very well for data, quite suited for tests

```
receipt:     Oz-Ware Purchase Invoice
date:        2007-08-06
customer:    Dorothy Gale
items:
    - part_no:   A4786
      descrip:   Water Bucket (Filled)
      price:     1.47
      quantity:  4
```

Testing framework: YAML

- Supports everything supported by XML and more
  - Calling internal service such as workflows
  - Arbitrary python code
- When added to the test section of the module descriptor, tests are rolled back by default
- New logging level `test` for showing test results

OpenERP YAML Syntax

- OpenERP supports specific custom types:
  - `!record {model: res_model, id: xml_id, context: {…} }`
  - `!python {model: res_model}`
  - `!workflow {model:res_model, action:signal, ref: xml_id}`
- And equivalents for all XML supported elements
  - `!act_window`
  - `!report`
  - `!function`
  - `!menuitem`
  - `!assert`
  - etc.

Simple YAML Example

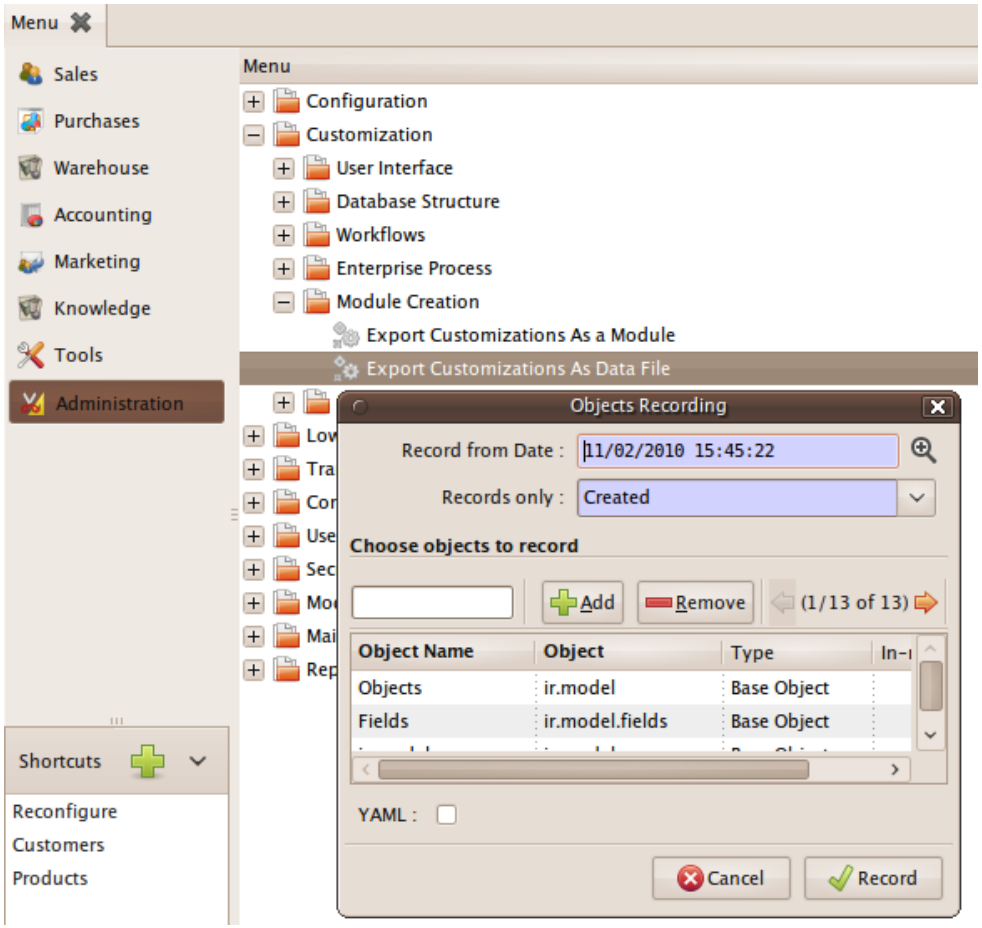
```
3  -
4    Create project 'OpenERP Training Programme'
5  -
6    !record {model: project.project, id: project_project_openerptrainingprogramme0}:
7      company_id: base.main_company
8      name: OpenERP Training Programme
9  -
10   Create task 'Technical Training' for this project
11 -
12   !record {model: project.task, id: project_task_technicaltraining0}:
13     date_start: '2010-05-31 11:48:38'
14     name: Technical Training
15     planned_hours: 30.0
16     project_id: project_project_openerptrainingprogramme0
17     remaining_hours: 30.0
18     state: draft
```

Advanced YAML Example

```
179 -
180   Make a work task entry 'Training on workflows' of 10 hours
181 -
182   !record {model: project.task, id: project_task_technicaltraining0, context:{'no_analytic_entry':True}}:
183     work_ids:
184       - date: '2010-06-02 15:05:24'
185         hours: 10.0
186         name: Training on workflows
187         user_id: base.user_root
188 -
189   Reevaluate the task
190 -
191   !python {model: project.task.reevaluate}: |
192     self.compute_hours(cr, uid, [ref("config_compute_remaining_0")], {"lang": "en_US",
193       "project_id": False, "tz": False, "active_model": "project.task", "search_default_project_id":
194       False, "search_default_user_id": 1, "search_default_current": 1, "active_ids":
195       [ref("project_task_technicaltraining0")], "active_id": ref("project_task_technicaltraining0"), })
196
197 -
198    Check for effective hours and remaining hours, remaining_hours must be 10 while planned_hours remains 30
199 -
200    !assert {model: project.task, id: project_task_technicaltraining0, severity: error, string: After... }
201        - planned_hours == 30
202        - remaining_hours == 10.0
203        - effective_hours == 30.0
```

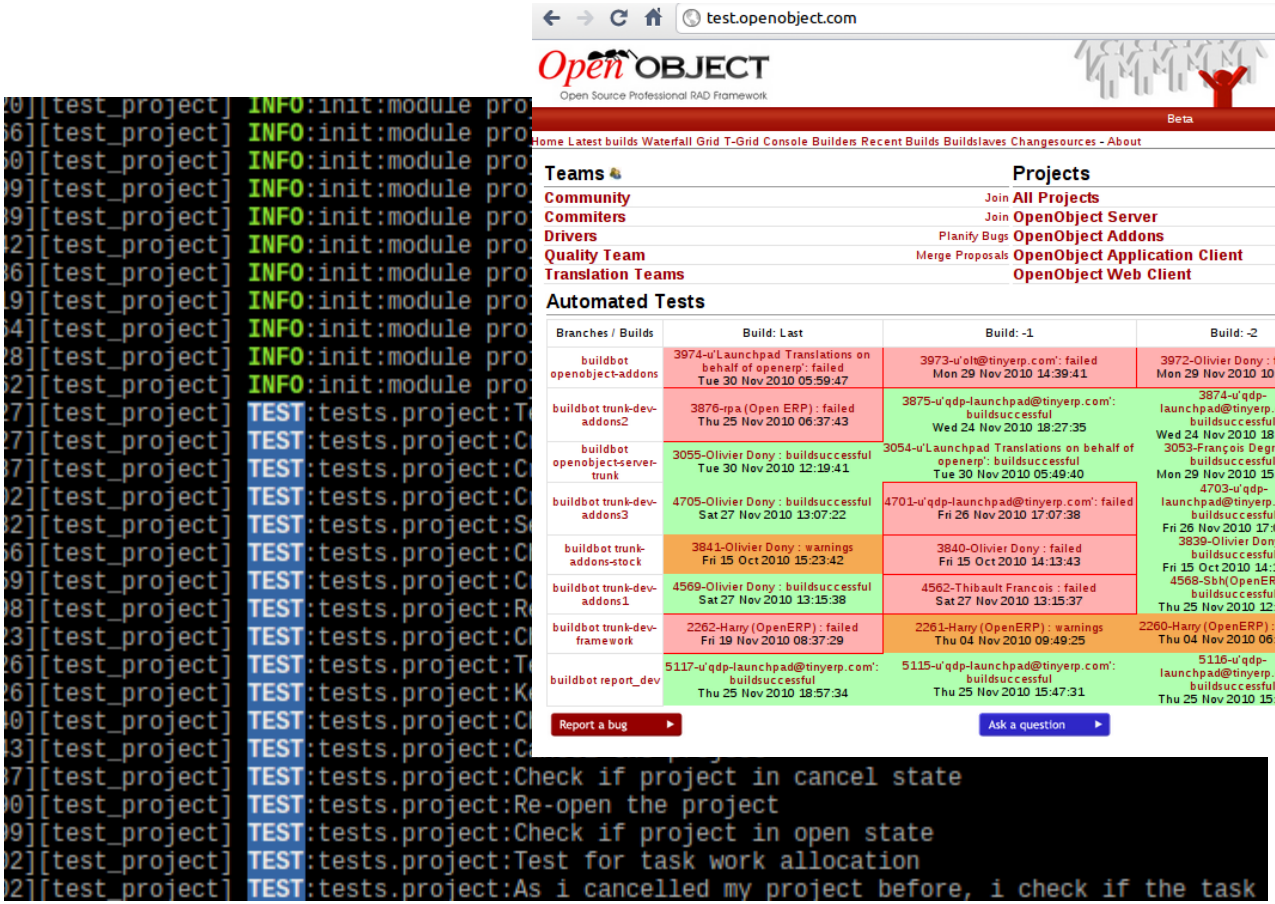
YAML: Scenario Export

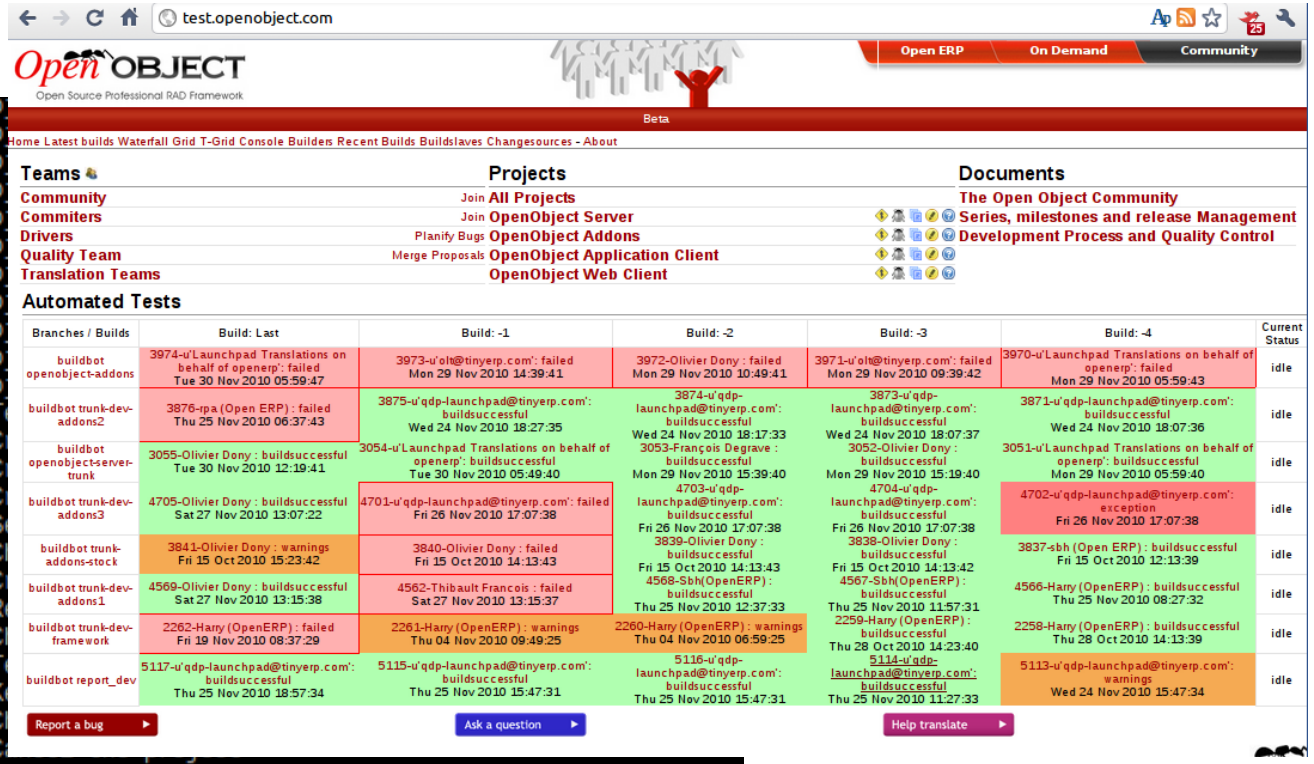
- `base_module_record` can now output YAML as well



YAML: Continuous Integration

- http://test.openobject.com (buildbot) builds each of our hundreds of YAML tests after each commit





Agenda

- Part 1
  - Short overview
  - Search views
  - Improved Tree views, aggregation, dynamic reports
  - Diagram views
  - Multi-company
- Part 2
  - Testing framework and continuous integration
  - Usability guidelines
  - New R&D Process
  - Web add-ons
  - Miscellaneous improvements and tips

Usability Guidelines

- Several Guidelines sections have been added to the online documentation (doc.openerp.com)
  - Coding Guidelines
  - YAML Guidelines
  - Usability Book (including guidelines)
- Usability guidelines are very important for 6.0!
- Needed to maintain coherent look and feel
- Help users directly locate what they need, when they need it, to be efficient *“Don't make me think”*
- Hide the complexity!

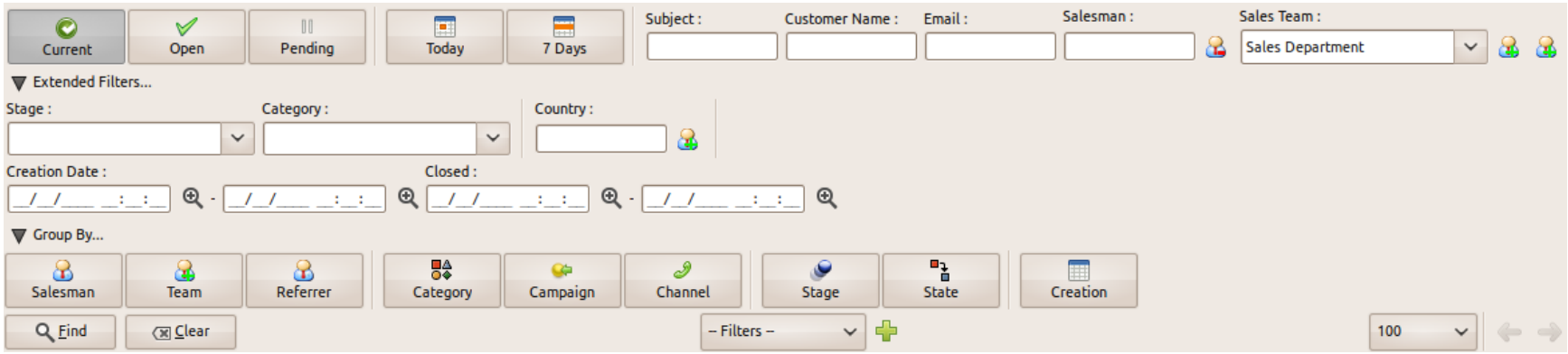
Usability Guidelines

- Integrate within Business Applications
  - Note: Groups and Access Rights are per Business Application (not per module)
- Keep menu structure “lean”, and in logical sequence
- Design Search Views
- Integrate/use the new configuration wizards
  - See new `res.config` and `res.config.installer` core classes
- Add Dashboard views and real-time Reports (group_by)

Usability: menu structure

- Most important items / First ones at top
- Reporting is at bottom
- Configuration is hidden (admin only)
- Use short, business-oriented names
- Example:
  - Main application
    - (Address book)
    - … other items ...
    - (Billing)
    - (Products)
    - Reporting
    - Configuration

Usability: Search Views



- Filters are on top-left, grouped by context, most important ones first
- Then search fields, same order as list columns
- Optional collapsed group for “Extended filters”
  - Organized by logical context
  - Add nested filter to toggle column visibility when useful
- Optional collapsed group for “Group-By”
  - Buttons organized by logical context
  - Usually: Persons, Document States, Places, Dates

OpenERP
OPEN SOURCE BUSINESS APPLICATIONS

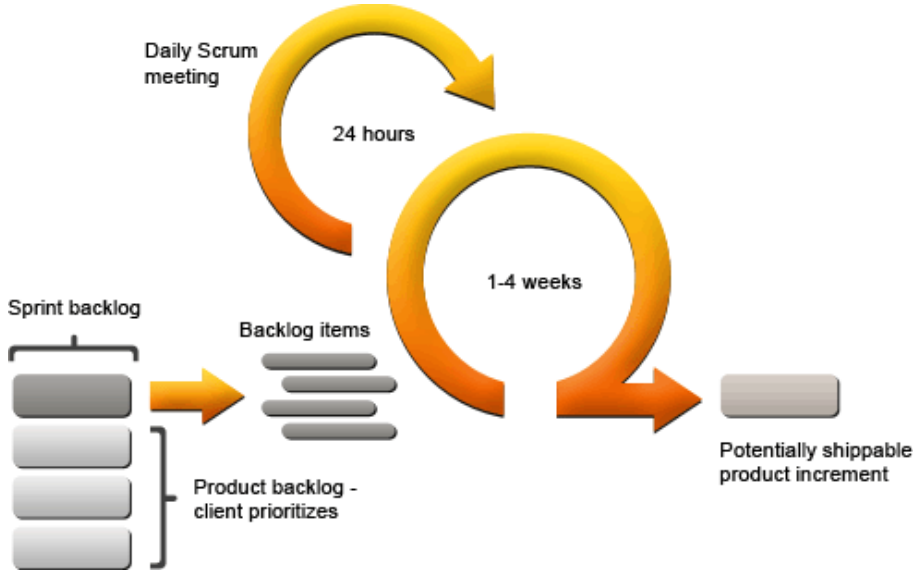
Agenda

- Part 1
  - Short overview
  - Search views
  - Improved Tree views, aggregation, dynamic reports
  - Diagram views
  - Multi-company
- Part 2
  - Testing framework and continuous integration
  - Usability guidelines
  - New R&D Process
  - Web add-ons
  - Miscellaneous improvements and tips

New R&D Processes

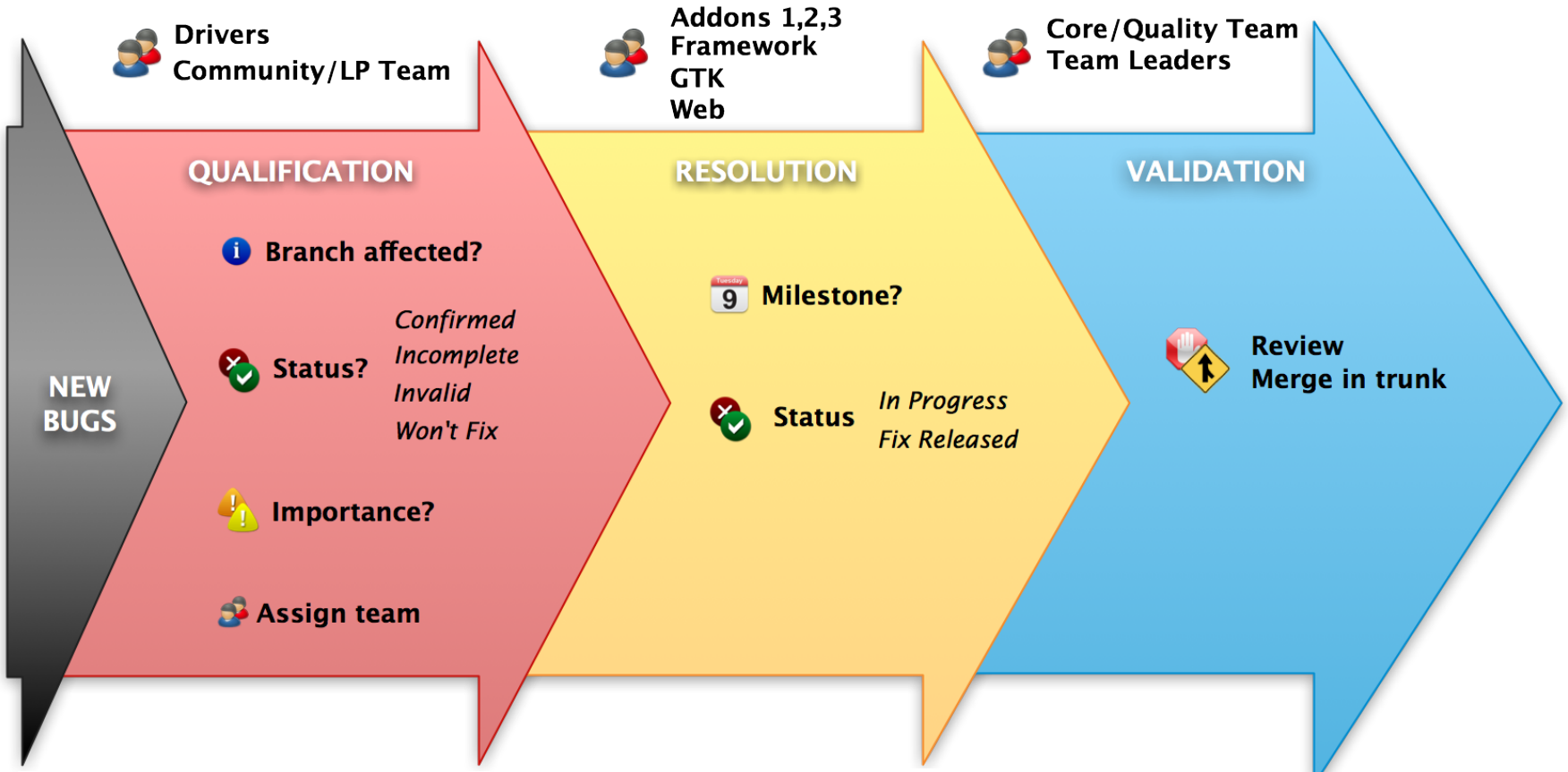
- Agile methodology for Product development: SCRUM
- Formal Bug management and Bug handling process
  - R&D teams distinct from Support/Maintenance teams
- Strong focus on
  - Quality
  - Peer Reviews
  - Continuous testing
  - Continuous improvement

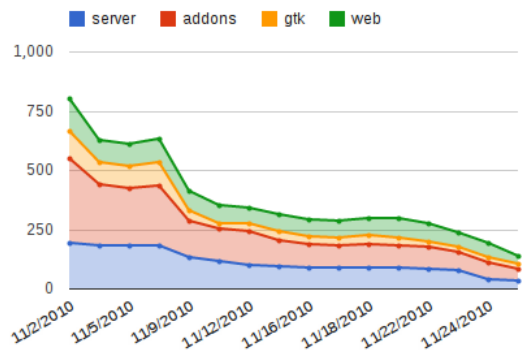
Product Development Process

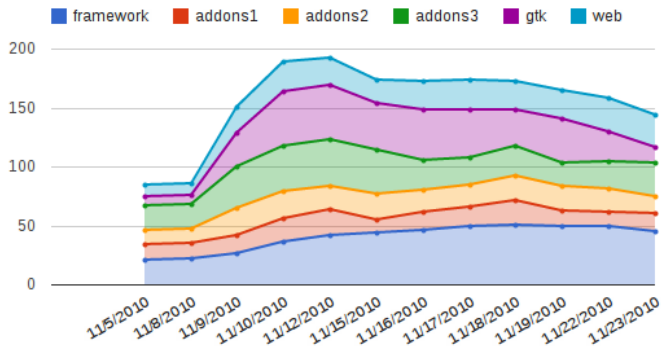


- Agile SCRUM sprints
- Sprint contents: Product backlog + Launchpad
- R&D Teams:
  - Web, GTK, Framework
  - Addons1, Addons2, Addons3
- All code reviewed by team leaders before end-of-sprint and merge into trunk

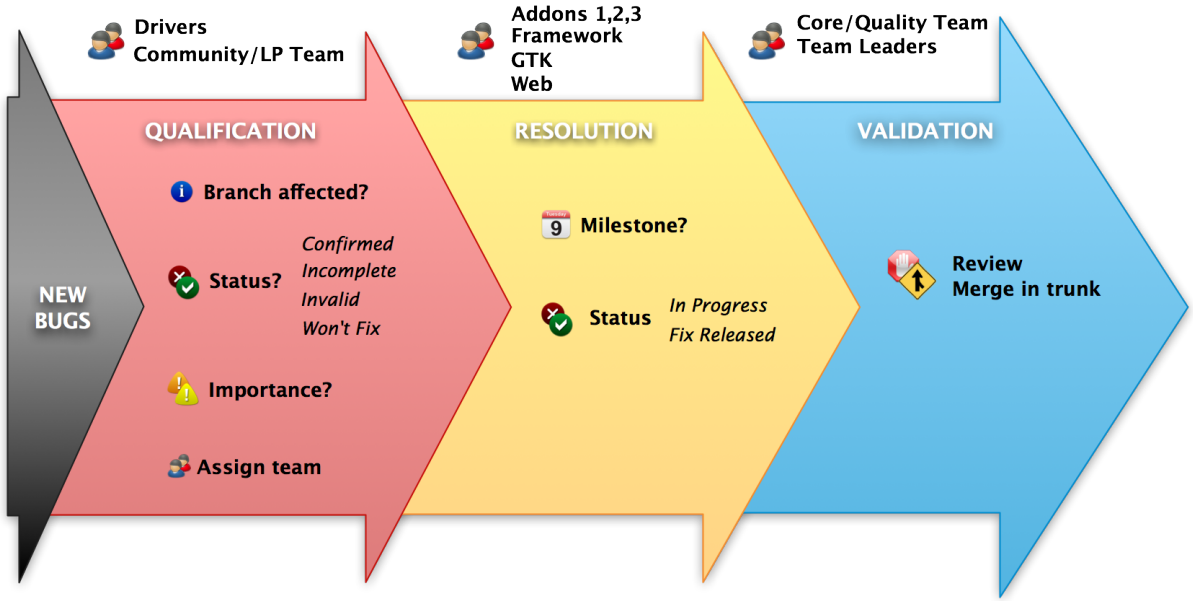
Bug Management Process

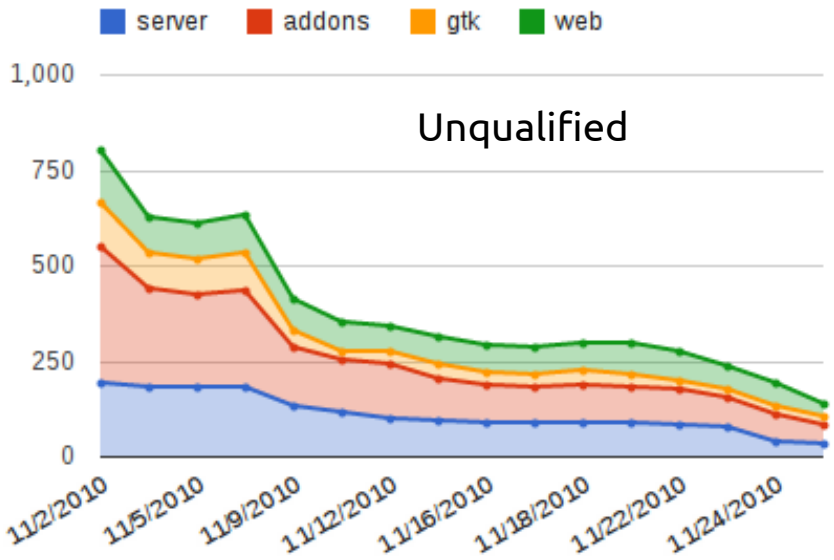


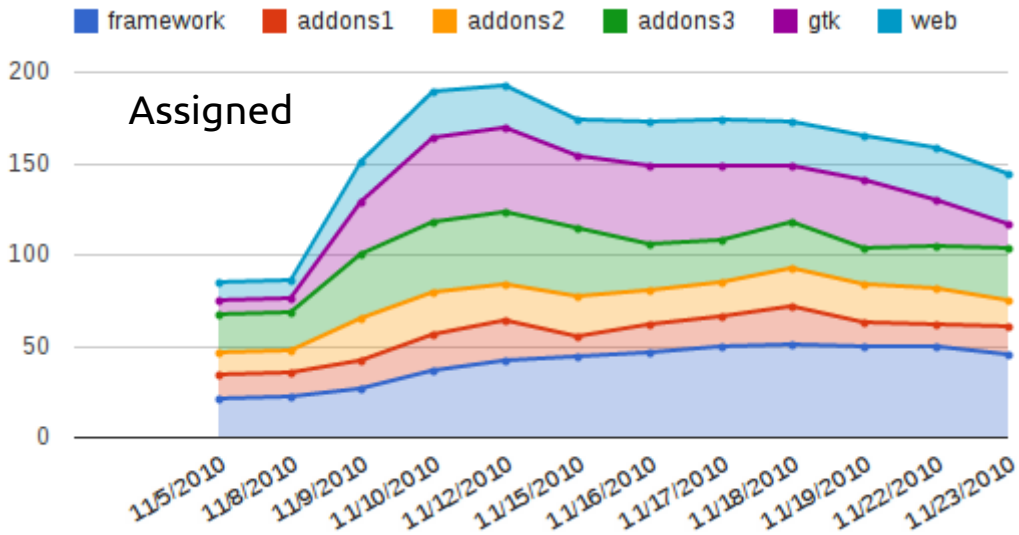




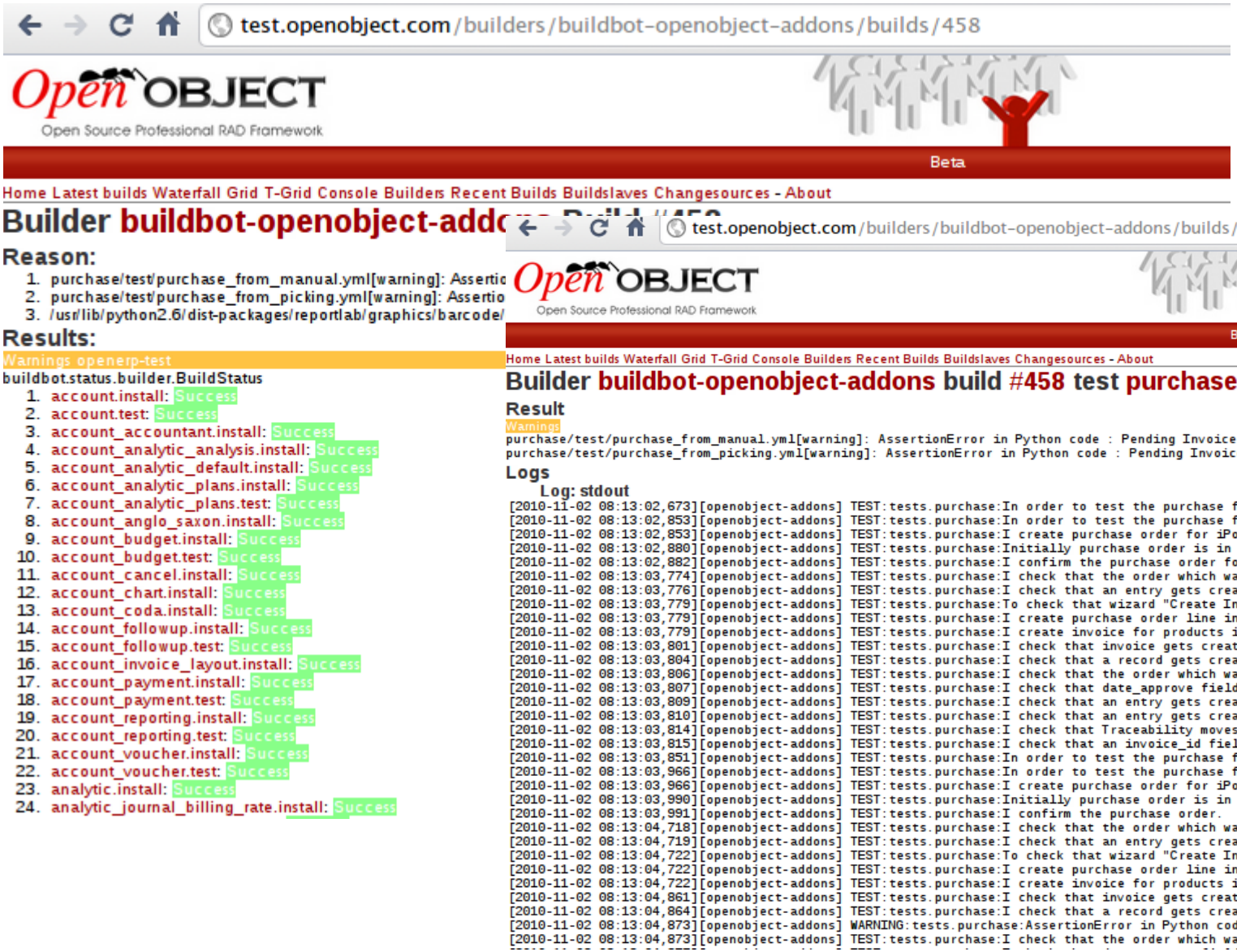
Bug Management Process







Continuous Integration



OpenERP
OPEN SOURCE BUSINESS APPLICATIONS

Agenda

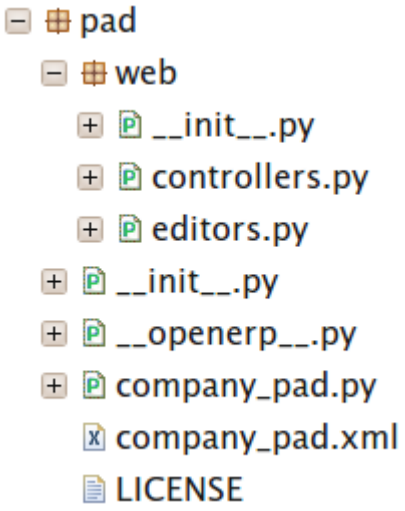
- Part 1
  - Short overview
  - Search views
  - Improved Tree views, aggregation, dynamic reports
  - Diagram views
  - Multi-company
- Part 2
  - Testing framework and continuous integration
  - Usability guidelines
  - New R&D Process
  - Web add-ons
  - Miscellaneous improvements and tips

Web Add-ons

- Web client was overhauled for v6.0: design and technology
- Web client may now have add-ons
- Web Add-ons are embedded in regular modules
- Web Add-ons may alter the default web client in the following areas:
  - Controllers
  - Templates
  - Widgets

Web Add-ons: structure

- Embedded within `web` directory inside module
- Easy deployment
- Inspired by server-side modules, but simplified
- `__openerp__.py`
  - name and dependencies, you may add your own metadata
- `__init__.py`
  - Python package, ensure that everything that needs to be loaded within web client is imported



Web-Addons: Controllers

- Inherit from either:
  - `openobject.controllers.BaseController`
  - `openerp.controllers.SecuredController`
    (for logged-in controllers)
- CherryPy exposition of methods, `@expose` shortcut
- `_cp_path` to define the controller's root
- Can Python-inherit from existing controller object to extend
- Set same `_cp_path` to override existing controller

Web Add-on: Controller Example

```python
class Etherpad(openerp.controllers.SecuredController):
    _cp_path = "/etherpad"

    @expose('json', methods=('POST',))
    def link(self, pad_name):
        params, data = TinyDict.split(cherrypy.session['params'])
        ctx = dict(rpc.session.context,
            default_res_model=params.model,
            default_res_id=params.id)

        pad_link = make_link(pad_name)
        attachment_id = rpc.RPCProxy('ir.attachment').create({
            'name': pad_name,
            'url': pad_link,
            }, ctx)
        return {
            'id': attachment_id,
            'name': pad_name,
            'url': pad_link}
```

Web Add-on: Templates Extension

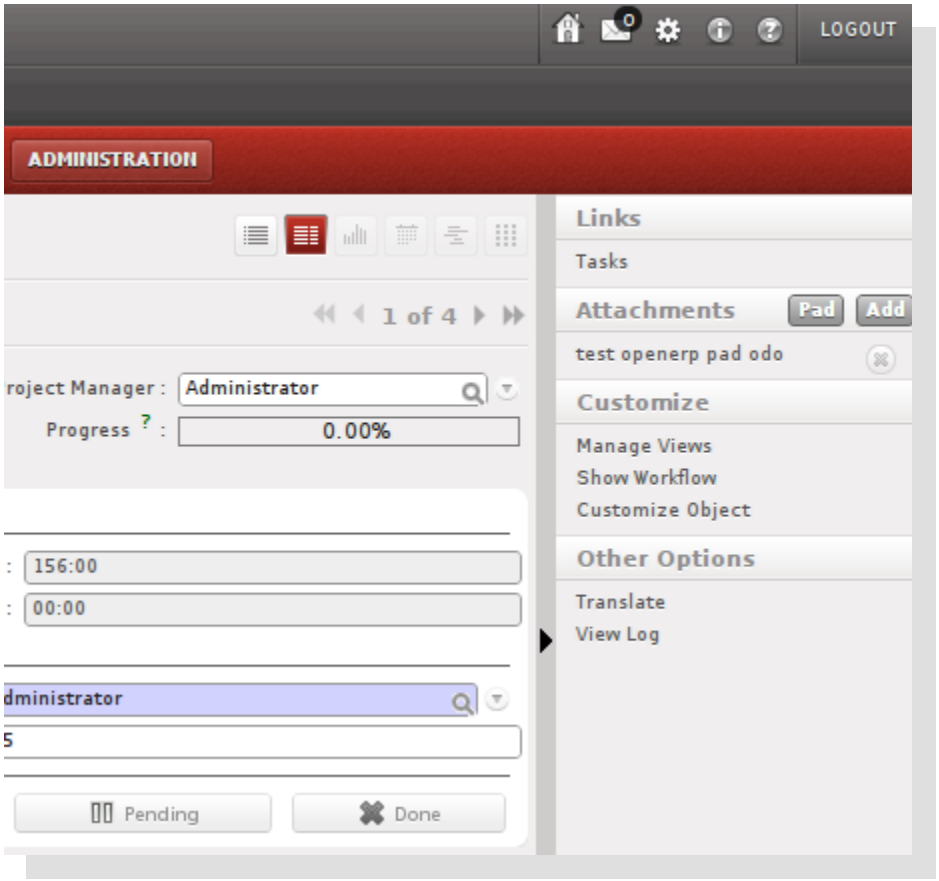
- Web client uses Mako-based templates
- Add-ons can register editor objects to do text manipulation of the “raw” mako code
- Example:

```
class SidebarTemplateEditor(openobject.templating.TemplateEditor):
    templates = ['/openerp/widgets/templates/sidebar.mako']
    ADD_ATTACHMENT_BUTTON = u'id="add-attachment"'

    def edit(self, template, template_text):
        output = super(SidebarTemplateEditor, self).edit(template, template_text)
        form_opener_insertion = output.index(
            '\n',
            output.index(self.ADD_ATTACHMENT_BUTTON)) + 1
        output = output[:form_opener_insertion] + \
            '''<a href="#" id="add-pad" class="button-a"
               style="right: 36px;"
               >${_("Pad")}</a>\n''' + \
            output[form_opener_insertion:]
        return output
```

Web Add-on: Example

- See the pad module for an example



HTML in views

- Apart from web add-ons, `<html>` may now be embedded in XML views
- Supported by web client only
- See `html_view` module for an example

Example:

```
<field name="arch" type="xml">
    <form string="Html Example">
        <html>
            <h2>This is Test Html Page </h2>
            <table>
            <tr>
                <td>
                    Name :
                </td>
                <td>
                    <field name="name"/>
                </td>
                <td>
                    Company :
                </td>
                <td>
                    <field name="comp_id"/>
                </td>
            </tr>
        </table>
            <field name="bank_ids"/>
        </html>
    </form>
</field>
```

Agenda

- Part 1
  - Short overview
  - Search views
  - Improved Tree views, aggregation, dynamic reports
  - Diagram views
  - Multi-company
- Part 2
  - Testing framework and continuous integration
  - Usability guidelines
  - New R&D Process
  - Web add-ons
  - Miscellaneous improvements and tips

Other Improvements

- Simpler __openerp__ structure: data, demo, test
- API:
  - `_defaults` may be literal
  - `_constraints`, `_sql_constraints` may be callables
- Security:
  - `safe_eval` replaced built-in `eval`
  - Private `_methods()` not callable via RPC
- Translations: improved and synchronized with LP
- Reporting engine more modular (e.g report_webkit)
- Performance improvements in RML reports
  - `<blockTable rowHeights="2cm" colWidths="11.0,7.0">`
  - Smarter translations, cached

Other Improvements

- Actions have new properties:
  - `help="Business description for this action"`
  - `multi=True`
- Simpler XML declaration for wizards actions
- ~~res.roles~~ => res.groups
- No more roles by default in workflows
- base_report_designer includes OOo plugin

- For more info, see:
  - Updated Technical Memento (Partner Preview available)
  - Technical Presentation Webinar
  - Online Documentation

Development Tips

- Improved views in Admin menus, with XML IDs
- 'View logs' shows XML ID now
- Debugging tooltips in GTK and Web
- new logging levels: `test`, `debug_rpc`, `debug_rpc_answer`, `debug_sql`
- command-line instance creation: `-d dbname -i base`
- Test reports
- `./openerp-server.py -i sale -d mytestdb --test-report-directory=/tmp`

This document is published under the **Creative Commons Attribution License 3.0 Unported** by OpenERP SA. Feel free to copy, share, or modify it.
For reference please see http://creativecommons.org/licenses/by/3.0/