Praktische Softwaretechnik

Prof. Dr. Dirk Riehle

Lecture 23/30 of 20.01.2010
<table>
<thead>
<tr>
<th>#</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Process Frameworks</td>
</tr>
<tr>
<td>2</td>
<td>Plan-Driven Software Development</td>
</tr>
<tr>
<td>3</td>
<td>Agile Methods (Scrum + Extreme Programming)</td>
</tr>
<tr>
<td>4</td>
<td>Open Source Software Development</td>
</tr>
<tr>
<td>5</td>
<td>Comparison of Frameworks</td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
</tbody>
</table>
The Three Main Process Frameworks

Plan-Driven

Agile Methods

Open Source
Plan-Driven Software Development

- Linear, phase-oriented process
  - Main goal is to minimize risk through careful upfront planning
  - Equates phases with activities
  - No iterations, just one pass
An Illustration of Plan-Driven Development

Picture credit: From http://www.megawoosh.com
Example Phases in Plan-Driven Development

- Project negotiations
- Requirements analysis
- Contract definition
- System analysis
- Software architecture design
- System design
- User interface design
- Implementation
- System test
- Acceptance test
- Handover
- Actual use
- Lawsuit

no feedback until after delivery
Agile Methods

- Agile methods is the name of a class of process frameworks
  - Extreme Programming, SCRUM, DSDM, Adaptive Software Development, Crystal, Feature-Driven Development, Pragmatic Programming, etc.
  - Unified by the recognition of a common philosophical base and joined in their rejection of the traditional life-cycle model
Agile Philosophy and Agile Manifesto

• Core agile method values
  – Individuals and interactions over processes and tools
  – Working software over comprehensive documentation
  – Customer collaboration over contract negotiation
  – Responding to change over following a plan

• Codified (and marketed) as the Agile Manifesto
  – See http://www.agilemanifesto.org
Two Example Agile Methods

Scrum
(mostly as process framework)

Extreme Programming (XP)
(mostly for development practices)
<table>
<thead>
<tr>
<th>User Stories</th>
<th>Sprint Planning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature Prioritization</td>
<td>Sprint Review</td>
</tr>
<tr>
<td>Release Planning</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Test First</td>
<td>Daily Release</td>
</tr>
<tr>
<td>Pair Programming</td>
<td>Refactoring</td>
</tr>
<tr>
<td>Continuous Integration</td>
<td>...</td>
</tr>
</tbody>
</table>
Scrum’s Development Process

- Succession of equal-length sprints (= short iterations)
- Intervention points are during planning and review
- Product owner is always available to answer questions

\[ \text{P} = \text{planning} \quad \text{E} = \text{engineering} \quad \text{R} = \text{review (or release or retrospective)} \]
The Significance of (Short) Iterations

- Short iterations lead to focus on high-value features first
- User feedback helps team steer product to meeting needs right
- Quality feedback helps team deliver high-quality software
- Feedback loop ensures problems and hence risk surfaces early
- Iterations help recognize and realize new innovative features
- Established well-worn rhythm is sustainable, avoids burnout
- Even with missed deadlines, partial functionality is better than none

Establishes shortest economically feasible feedback loop
Structure of an Exemplary Sprint

day 1
Thursday
- sprint planning
  - product backlog development
  - software development
- daily scrum
- noon
- start

day 2
Friday
- product backlog development
- software development
- daily scrum
- noon
- start

day 3
Monday
- product backlog development
- next sprint preparation
- software development
- daily scrum
- noon
- start

...
Test-Driven Development

• Test-driven development (abbrev. TDD)
  – Is a core process for programming activities
  – Promises minimal fluctuations in development speed
  – Takes the stress out of refactoring your code base
  – Considered a high-end practice (requires too much discipline for many)
  – Requires significant experience with ancillary tools and techniques

• First described by Kent Beck in [Beck 2003]
TDD Follows the XP Mantra

Make it run

Make it right

Make it fast
The Smallest Possible (Believable) Process

... Description n+2
Description n+1
Description n
Description n-1
Description n-2
...

... Test Suite n
Test Suite n-1
Test Suite n-2
...

Implementation n
Implementation n-1
Implementation n-2
...

on to next feature
enhance test suite
The Promise of Test Driven Development

- You can release after every feature implementation
- The likelihood of failure or bugs is as minimal as possible
- The product grows incrementally and steadily
- The code is of pristine (and lean) quality

- TDD promises minimal fluctuations in development speed
TDD Subprocesses

1. Translate description into test suite
2. Implement feature to fulfill (“green-bar”) test suite
3. Revise test suite from new domain and implementation insights
4. Refactor test suite to keep code healthy
5. Refactor implementation to keep design and code healthy
6. Exit when test suite is complete and all tests succeed
TDD is a Complex Adaptive System

Two simple rules

<table>
<thead>
<tr>
<th>only write new code when a test fails</th>
</tr>
</thead>
<tbody>
<tr>
<td>eliminate waste</td>
</tr>
</tbody>
</table>

deliver

clean code that works
Definition of Open Source (Abbreviated)

• Open source software is software that is
  – available in source code form
  – can be freely used and modified
  – can be freely redistributed
  – cf. four freedoms of software

• The open source initiative
  – maintains the definition and trademark
  – approves open source licenses
Some Open Source Projects

- Linux
- Apache
- SugarCRM
- OpenOffice.org
- MySQL
- JasperSoft
- Eclipse
- PostgreSQL
- Debian
- Python
Free/Libre/Open Source Software

• The free software movement
  – Initiated (and still led) by Richard Stallman (MIT) in the 1980 to free software (from being closed)
  – Free software philosophy summarized as “free as in 'free speech' not as in 'free beer’”

• Free Software Foundation
  – http://www.fsf.org
  – U.S. 501(c) non-profit organization

• The open source movement
  – Eric Raymond: “Cathedral and the Bazaar” --- describes open source as development method
  – Formalized 1998 to address perceived anti-commerce bias of “free software”

• Open Source Initiative
  – http://www.opensource.org
  – U.S. 501(c) non-profit organization

FLOSS = Free/Libre/Open Source Software
Economic Impact of Open Source

• Commercial use
  - Gartner: “By 2012, more than 90 percent of enterprises will use open source […]” [1]
  - By and large, open source has gone mainstream, is just a product like any other
  - Today, formal open source adoption strategy elusive and TCO gains unclear
  - Open source dominates software-as-a-service and in startups

• Packaged software market
  - In 2006, open source held $1.8B / $235B = 0.8% of market revenue [2] [3]
  - IDC: Open source revenue will reach $5.8B by 2011 (26% CAGR 2006-11) [2]

Current and Projected Growth Pattern


(This is a guesstimate; not drawn to scale)
## Types of Open Source Projects

<table>
<thead>
<tr>
<th>Single product or product line</th>
<th>Community-owned</th>
<th>Single or dominant proprietor</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Community Open Source</strong></td>
<td>Community-owned</td>
<td><strong>Commercial Open Source</strong></td>
</tr>
<tr>
<td>(e.g. Linux, TikiWiki)</td>
<td></td>
<td>(e.g. MySQL, Jasper)</td>
</tr>
<tr>
<td><strong>Community Distribution</strong></td>
<td>Community-owned</td>
<td><strong>Commercial Distribution</strong></td>
</tr>
<tr>
<td>(e.g. Debian)</td>
<td></td>
<td>(e.g. RHEL, SLES)</td>
</tr>
<tr>
<td><strong>Multi-product assembly (“stack”)</strong></td>
<td>Community-owned</td>
<td><strong>Commercial Distribution</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(e.g. Debian)</td>
</tr>
</tbody>
</table>
Time-line of Open Source

1991: Linux project started

1998: Open Source Initiative founded

Traditional Community Open Source

1999: Apache Software Foundation founded

Managed Community Open Source

2004: Eclipse Foundation founded

1995: MySQL AB founded

2001: MySQL AB funded

Single Vendor ("Commercial") Open Source
Open Collaboration [1]

Egalitarian

Meritocratic

Self-organizing

<table>
<thead>
<tr>
<th>Traditional Work</th>
<th>Open Collaboration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hierarchical</td>
<td>Egalitarian</td>
</tr>
<tr>
<td>- Closed and hidden silos</td>
<td>- Open for contribution</td>
</tr>
<tr>
<td>- Assigned to project</td>
<td>- Everyone can contribute</td>
</tr>
<tr>
<td>Status-driven</td>
<td>Meritocratic</td>
</tr>
<tr>
<td>- Public and private discussions</td>
<td>- Public discussion process</td>
</tr>
<tr>
<td>- Hierarchical status decides</td>
<td>- Decisions based on merit</td>
</tr>
<tr>
<td>Assigned tasks</td>
<td>Self-organizing</td>
</tr>
<tr>
<td>- Prescribed process</td>
<td>- People find their own process</td>
</tr>
<tr>
<td>- Prescribed jobs</td>
<td>- People find their best project</td>
</tr>
</tbody>
</table>
Roles in Traditional Processes

- **Product Management**
  - Product Manager

- **Research and Development**
  - Development Manager
  - Developer
  - …
  - Developer

- **Quality Assurance**
  - Tester
Artifacts in Traditional Processes

PM

Product Manager

Requirements Document

R&D

Development Manager

Time Table

Developer

Program Code

QA

Tester

Reports and Test Code
Committers, Contributors, Community

- Committers
  - Main developers
  - With access rights

- Contributors
  - Casual developers
  - Submit patches

- (User) community
  - Provide bug reports
  - Provide feedback
PostgreSQL Product Management

• Who defines features? [1]
  – Patches
  – Scratching one's itch
  – Individual developer initiatives
  – Company-driven contributions
  – User polls on project home page

• Product management practices
  – Leadership by core team
  – Maintenance of a todo list [2]
  – No prioritization: “Pick your feature”

PostgreSQL Development

PostgreSQL Official releases
Versions 8.x, 7.4

Core Team [1] Major Contributors

EnterpriseDB
2+1
Product

Command Prompt
0+3
Product

ACME
0/1+0/1
Product

Anon B. Coder

Some Fix

PostgreSQL Quality Assurance

PostgreSQL
Official releases
Versions 8.x, 7.4

Core Team
Major Contributors

Patch Review
Round Robin
CommitFest

Anon B. Customer
Developer
Project

Anon B. Company
Developer
Product

Anon B. Coder
Developer
Project
<table>
<thead>
<tr>
<th>new application domain</th>
<th>no</th>
<th>yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>no</td>
<td>plan-driven agile methods open source</td>
<td>agile methods open source</td>
</tr>
<tr>
<td>yes</td>
<td>plan-driven open source</td>
<td>agile methods open source</td>
</tr>
</tbody>
</table>
Questions? Feedback!

http://dirkriehle.com - dirk.riehle@cs.fau.de - @dirkriehle