What colours is your backlog?

Philippe Kruchten
@pbpk
Philippe Kruchten, Ph.D., P.Eng., CSDP

Professor of Software Engineering
NSERC Chair in Design Engineering
Department of Electrical and Computer Engineering
University of British Columbia
Vancouver, BC Canada
pbk@ece.ubc.ca

Founder and president
Kruchten Engineering Services Ltd
Vancouver, BC Canada
philippe@kruchten.com
The Backlog

Feature Requests

Priority
Sprint Planning

Feature Requests → Priority → Sprint Backlog
But what is in your backlog?
Painting your backlog

<table>
<thead>
<tr>
<th>Features</th>
<th>Architecture infrastructure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defects</td>
<td>Technical Debt</td>
</tr>
</tbody>
</table>
Time-box
Time-box

Staff

Time
Time-box

Staff

Work (or Effort)

Time
Time-box

Staff

Work (≈Cost)

Time
Time-box

better than

Brooks, Mythical Man-Month, 1975
Boehm, COCOMO, 1981
Time-boxes: Releases

Release 1  R2  R3  R4

Time
Time-boxes: Iterations (sprints)
Features

Intent
Features
Features
Features
Features & Value

Utils
Maximizing value

Highest value first
Ignore time
Cost?
Value = Cost?

Points

Only for simplest cases
Value /≠ Cost

Value

Cost

5 $1

7 $2

5 $5

4 $5

4 $5

3 $5

3 $5

2 $5

6 $8

12 $15
Value and Cost

• Value: to the business (the users, the customers, the public, etc.)
• Cost: to design, develop, manufacture, deploy, maintain

• Simple system, stable architecture, many small features:
  – Statistically value aligns to cost
• Large, complex, novel systems?
Efficiency vs. Effectiveness

**Efficiency**
- relationship between the output in terms of goods, services or other results and the resources used to produce them

**Effectiveness**
- relationship between the intended impact and the actual impact of an activity

| Cost | Value |
Cost will impose the limit
What colour is your backlog?

(so far)
Invisible Features

12 $15

12 $10

? $5
Features and ...
Invisible features

• Architecture
• Infrastructure
• Frameworks
• ....
Architecture: Value and Cost

- Architecture has no (or little) externally visible "customer value"
- Iteration planning (backlog) is driven solely (?) by "customer value"
- YAGNI, BUFD, Metaphor,...
- "Last responsible moment!"
- We’ll refactor it later!
- *Ergo*: architectural activities are not given proper attention
- *Ergo*: large technical debts
Value

\[ \sum = 186 \text{ utils} \]
Value reallocated to architecture

\[ \sum = 186 \text{ utils} \]
Features
Features
Visible & Invisible Features
Time-box
Time-box with Buffer
What colour is your backlog?

(so far)
Defects

• Defect = Feature with negative value

• Fix (defect) has a positive cost (= work)

• Time/place of discovery
  – Inside development (in-house, in process)
  – Outside development in a released product (escaped defects)
Escaped Defect has Negative Value

Perfect product

Imperfect product

Defect
Buffer for in-process defects
Fixing a Defect has a Cost

• Defects have both value and cost
• Value of fixing a defect = −Value of the defect
• Cost of fixing a defect (estimated)

• Defects have dependencies
  – Defect fix depend on invisible feature
  – Visible feature depending on a fix
Visible and Invisible Features
Visible & Invisible Features + Defects fixing
What colour is your backlog?

(so far)
Technical Debt

- Concept introduced by Ward Cunningham
- Often mentioned, rarely studied
- All experienced SW developers “feel” it.
- Drags long-lived projects and products down
- Friction

Cunningham, OOPSLA 1992
Origin of the metaphor

• Ward Cunningham, at OOPSLA 1992

“Shipping first time code is like going into debt. A little debt speeds development so long as it is paid back promptly with a rewrite...
The danger occurs when the debt is not repaid. Every minute spent on not-quite-right code counts as interest on that debt. Entire engineering organizations can be brought to a stand-still under the debt load of an unconsolidated implementation, object-oriented or otherwise.”

Cunningham, OOPSLA 1992
Technical Debt Definition (2013)

• A design or construction approach that is expedient in the short term, but that creates a technical context in which the same work will cost more to do later than it would cost to do now (including increased cost over time).

McConnell 2013
Technical Debt (S. McConnell)

• Implemented features (visible and invisible) = assets = non-debt
• Type 1: unintentional, non-strategic; poor design decisions, poor coding
• Type 2: intentional and strategic: optimize for the present, not for the future.
  – 2.A short-term: paid off quickly (refactorings, etc.)
    • Large chunks: easy to track
    • Many small bits: cannot track
  – 2.B long-term

McConnell 2007
Technical Debt (M. Fowler)

<table>
<thead>
<tr>
<th>Reckless</th>
<th>Prudent</th>
</tr>
</thead>
<tbody>
<tr>
<td>“We don’t have time for design”</td>
<td>“We must ship now and deal with consequences”</td>
</tr>
<tr>
<td>Deliberate</td>
<td></td>
</tr>
<tr>
<td>“What’s Layering?”</td>
<td>“Now we know how we should have done it”</td>
</tr>
</tbody>
</table>

Fowler 2009, 2010
Technical Debt (1)
Technical Debt (2)

12

12

a

$15

$5

b

$16

$3

8

8

$25

$27

8

8

$28

$10

$8
Technical Debt (3)
Interests

• In presence of technical debt:
  Cost of adding new features is higher

• When repaying (fixing), additional cost for retrofitting already implemented features

• Technical debt not repaid => lead to increased cost, forever

• Cost of fixing increases over time

M. Fowler
TD litmus test

- If you are not incurring any interest, then it probably is not a debt

McConnell 2013
Technical debt landscape

- New features
- Additional functionality
- Architectural debt
  - Test debt
- Structural debt
- Documentation debt
- Code complexity
  - Code smells
- Coding style violations
- Low internal quality
- Code
- Visible
  - Defects
  - Low external quality
  - Evolution issues: evolvability
- Quality issues: maintainability

Kruchten et al 2012
Technical debt

• Not just crappy code: wise investment

• Depends on the future

“Technical futures”
Repaying debt

• What to repay?

• When to repay?
Tackling Technical Debt

Attitudes and approaches found:
1. Ignorance is bliss
2. The elephant in the room
3. Big scary $$$$ numbers
4. Five star ranking
5. We’re agile, so we are immune!
6. Constant reduction
7. Reduction iterations (sprints)
Buffer for debt repayment

- Simple work
- Estimate uncertainties
- Defect correction
- Debt Repayment
Colours in your Backlog

<table>
<thead>
<tr>
<th>Positive Value</th>
<th>Visible Feature</th>
<th>Hidden, architectural feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negative Value</td>
<td>Visible defect</td>
<td>Technical Debt</td>
</tr>
</tbody>
</table>
Visible & Invisible Features + Defects fixing + Technical Debt payment
Tensions

Product manager

Visible Feature

Hidden, architectural feature

Visible defect

Technical Debt

Architects

Nobody?
Tools !?
<table>
<thead>
<tr>
<th>#</th>
<th>Summary</th>
<th>Cost</th>
<th>Value</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>Tell Anonymous Users the benefits of registering</td>
<td>Low Cost</td>
<td>High Value</td>
<td></td>
</tr>
<tr>
<td>#103</td>
<td>Attachments should be possible for all comments, including threaded replies</td>
<td>Medium Cost</td>
<td>High Value</td>
<td></td>
</tr>
<tr>
<td>#97</td>
<td>Legal notices</td>
<td>Low Cost</td>
<td>Medium Value</td>
<td></td>
</tr>
<tr>
<td>#100</td>
<td>Per-list unsubscribe for mailing-list-only users</td>
<td>Medium Cost</td>
<td>Medium Value</td>
<td></td>
</tr>
<tr>
<td>#50</td>
<td>Group member list: &quot;edit details&quot; for someone's membership</td>
<td>Low Cost</td>
<td>Medium Value</td>
<td></td>
</tr>
<tr>
<td>#96</td>
<td>Chapters (possibly networks?) map</td>
<td>Medium Cost</td>
<td>Low Value</td>
<td></td>
</tr>
<tr>
<td>#102</td>
<td>Topics should show icon of some sort to indicate attachments</td>
<td>Low Cost</td>
<td>Low Value</td>
<td></td>
</tr>
</tbody>
</table>

**1.0 RC1 - Release Candidate 1**

(14 matches)

<table>
<thead>
<tr>
<th>Ticket</th>
<th>Summary</th>
<th>Cost</th>
<th>Value</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>#49</td>
<td>switch to email-based usernames</td>
<td>High Cost</td>
<td>High Value</td>
<td>joshuagomer</td>
</tr>
<tr>
<td>#54</td>
<td>Who's Online listing</td>
<td>Medium Cost</td>
<td>High Value</td>
<td>francis</td>
</tr>
<tr>
<td>#55</td>
<td>National Office content</td>
<td>Medium Cost</td>
<td>High Value</td>
<td></td>
</tr>
<tr>
<td>#58</td>
<td>chapter vs network</td>
<td>Medium Cost</td>
<td>High Value</td>
<td>joshuagomer</td>
</tr>
<tr>
<td>#61</td>
<td>Intuitive combinations of group visibility / privacy in UI</td>
<td>Medium Cost</td>
<td>High Value</td>
<td></td>
</tr>
<tr>
<td>#93</td>
<td>Individual anonymous users should be able to sign up to mailing lists</td>
<td>Medium Cost</td>
<td>High Value</td>
<td></td>
</tr>
<tr>
<td>#48</td>
<td>network_new_member cannot use dropdown to list members</td>
<td>Medium Cost</td>
<td>High Value</td>
<td></td>
</tr>
<tr>
<td>#81</td>
<td>Prevent &quot;private&quot; networks</td>
<td>Medium Cost</td>
<td>High Value</td>
<td></td>
</tr>
<tr>
<td>#98</td>
<td>Group creator (particularly networks) need not also be a member</td>
<td>Medium Cost</td>
<td>High Value</td>
<td></td>
</tr>
<tr>
<td>#21</td>
<td>Verify email accounts automatically</td>
<td>Medium Cost</td>
<td>High Value</td>
<td></td>
</tr>
<tr>
<td>#22</td>
<td>Multiple levels of membership in a group</td>
<td>Medium Cost</td>
<td>Medium Value</td>
<td></td>
</tr>
<tr>
<td>#23</td>
<td>Groups should have &quot;former members&quot; to handle involvement history</td>
<td>Medium Cost</td>
<td>Medium Value</td>
<td></td>
</tr>
<tr>
<td>#56</td>
<td>Suggested communities</td>
<td>Medium Cost</td>
<td>Medium Value</td>
<td></td>
</tr>
<tr>
<td>#60</td>
<td>Notifications for group invitations / requests</td>
<td>Low Cost</td>
<td>Medium Value</td>
<td></td>
</tr>
</tbody>
</table>

**None**

(30 matches)

<table>
<thead>
<tr>
<th>Ticket</th>
<th>Summary</th>
<th>Cost</th>
<th>Value</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>#64</td>
<td>fire and forget URL for signing up email addresses to the main list</td>
<td>Medium Cost</td>
<td>High Value</td>
<td></td>
</tr>
<tr>
<td>#67</td>
<td>topic-creation preview</td>
<td>Medium Cost</td>
<td>High Value</td>
<td></td>
</tr>
<tr>
<td>#99</td>
<td>feedback system</td>
<td>Low Cost</td>
<td>High Value</td>
<td></td>
</tr>
<tr>
<td>#68</td>
<td>too many notices!</td>
<td>Low Cost</td>
<td>High Value</td>
<td></td>
</tr>
<tr>
<td>#78</td>
<td>Clicking on a tag causes an error</td>
<td>Low Cost</td>
<td>High Value</td>
<td></td>
</tr>
<tr>
<td>#90</td>
<td>private-messaging 'to' input very rough</td>
<td>Medium Cost</td>
<td>High Value</td>
<td>benbest</td>
</tr>
<tr>
<td>#91</td>
<td>password strength issue</td>
<td>Low Cost</td>
<td>High Value</td>
<td></td>
</tr>
</tbody>
</table>
Kanban

Each list is a pane of issues. The issues can be dragged and dropped onto other panes based on roles and permissions settings.

<table>
<thead>
<tr>
<th>User</th>
<th>Active</th>
<th>Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chris Nicola</td>
<td>#20 - Feature - Accounting Manager - David's changes in commission report</td>
<td>(No issues)</td>
</tr>
<tr>
<td></td>
<td>#21 - Feature - FSM Data Adapter - Create UI for loading cash transactions</td>
<td>(No issues)</td>
</tr>
<tr>
<td></td>
<td>#22 - Feature - General Tasks - Environment Setup</td>
<td>(No issues)</td>
</tr>
<tr>
<td></td>
<td>#23 - Feature - Insurance Manager - Display escape field in policy list</td>
<td>(No issues)</td>
</tr>
<tr>
<td>Adam Dymitrak</td>
<td>#24 - Feature - FSM Data Adapter - Evaluate Sales REDEP vs CRM data migration</td>
<td>(No issues)</td>
</tr>
<tr>
<td>Admin Stator</td>
<td>#25 - Bug - Unsorted - Test Bug for Kanban Board</td>
<td>(No issues)</td>
</tr>
<tr>
<td>Jennifer Keates</td>
<td>#26 - Technical Debt - Unsorted - Test Technical Debt for Kanban Board</td>
<td>(No issues)</td>
</tr>
<tr>
<td></td>
<td>#27 - Feature - FSM Data Adapter - Test technical debt for Kanban Board</td>
<td>(No issues)</td>
</tr>
<tr>
<td>Lee Tipton - Aylmer</td>
<td>#28 - Architecture - Unsorted - Test Architecture for Kanban Board</td>
<td>(No issues)</td>
</tr>
<tr>
<td>Paul Newton</td>
<td>#29 - Feature - Database - Client List Table</td>
<td>(No issues)</td>
</tr>
<tr>
<td>Richard Harris</td>
<td>(No issues)</td>
<td>(No issues)</td>
</tr>
<tr>
<td>Tim Low</td>
<td>(No issues)</td>
<td>(No issues)</td>
</tr>
</tbody>
</table>

Based on Redmine, by Chris Nicola
Risks & Uncertainties

Rules of thumb:
... facing uncertainty:
• Green stuff: move up
  – Defer
• Yellow stuff: move down
  – Experiment now

Karl Wiegers, 1999
Kruchten, 1998
Key message(s)

• Having multiple repositories of things to do, managed by multiple or different people, based on different criteria is a bad idea.
• It leads to delays, frustrations, accumulated technical debt, reduced velocity, distrust....
• Manage all colours together
• Value is different than cost
Tensions

Product manager

- Visible Feature
- Visible defect

Architects

- Hidden, architectural feature
- Technical Debt

Customer Support

Nobody?
Agility

• Lead to a shared mental model of the real state of the project
• Common understanding of the nature and extent of commitments
• Scale
Manage them all together
References

References


Other pointers

http://techdebt.org

http://www.ontechnicaldebt.com/

@OnTechnicalDebt