Managing Technical Debt
Practical Decision-Making and Its Business Relevance

Dr. Christof Ebert
Vector Consulting Services

OOP 2013, Munich, Germany
Dr. Christof Ebert is managing director at Vector Consulting Services. He supports clients around the world to improve product strategy and product development and to manage organizational changes. Prior to that, he held global management positions for fifteen years at Alcatel-Lucent.

Dr. Ebert serves on a number of advisory and industry bodies. An internationally renowned keynote speaker, he teaches at the University of Stuttgart and authored several books including his most recent book “Global Software and IT” published by Wiley.

christof.ebert@vector.com
www.vector.com/consulting
Vector Consulting Services

- offers a comprehensive consulting, tools and training portfolio for optimizing technical product development
- serves industries such as automotive, aviation, IT and telecom, energy and environment, medical and railway
- with clients such as Accenture, Audi, BMW, Bombardier, Bosch, Daimler, Denso, ESA, Hyundai, IBM, JCI, MAN, Porsche, SK Innovation, Siemens, Telefonica, Thales, Vodafone, Zeiss
- is supporting clients on efficiency improvement, processes, tools, organizational change management
- as a group serves companies across the world with over 1100 employees and sales of well over 200 Mio € pa
- [www.vector.com/consulting](http://www.vector.com/consulting)

© 2013 Dr. Christof Ebert, Vector Consulting Services GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector. V 2.0. 2013-01-03.
Principles: Accruing Technical Debt

Practices: 40 Ways to Leave Your Debt

Potentials: Managing the Debt
Technical Debt: A Historical Perspective

“The eventual consequences of poor engineering of a software product for short-term benefits that make the same work cost more to do later.”
— Christof Ebert, 2007

“As an evolving program is continually changed, its complexity, reflecting deteriorating structure, increases unless work is done to maintain or reduce it.”
— Meir Manny Lehman, 1980

“Technical debt accumulates to over 3.60 US$ per line of code.”
— CAST, Bill Curtis, 2011

“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.”
— Ward Cunningham, 1992

“Accidental complexity relates to problems that we create on our own and which can be fixed.”
— Fred Brooks, 1986
Technical Debt: An Engineering Perspective

- **Features**
  - Value
  - Cost
- **Architecture**
  - Quality attributes

- **Defects**
  - Overengineering

- **Technical Debt**
  - Careful vs. Careless
  - Aware vs. Unaware

"Visible" vs. "Invisible"

Technical Debt: A Managerial Perspective

- **Mergers and acquisitions**
  Is the company and its products viable and long-term profitable?

- **Due diligence**
  What is the risk of this product? How to calculate the real options?

- **Portfolio management**
  Shall we continue or stop this product?

- **Life-cycle cost**
  What is the cost of ownership?

---

Managing technical debt

Strategy → Concept → Market entry → Development → Evolution

Development paradigms, e.g. Agile
Technical Debt: An Educational Perspective

Source: http://www.sei.cmu.edu/architecture/tools/hardchoices/
Accruing Unmanaged Debts Will Strangulate Your Product

<table>
<thead>
<tr>
<th>Accruing Financial Debt</th>
<th>Accruing Technical Debt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Borrow money against a future date</td>
<td>Borrow money and time against a future date</td>
</tr>
<tr>
<td>More debt to pay interest rates</td>
<td>Complexity trap</td>
</tr>
<tr>
<td>“Virtual debt” by not having the best interest rate</td>
<td>Not making savings against time where possible</td>
</tr>
<tr>
<td>Low inventory turnover</td>
<td>Partially done work accumulates</td>
</tr>
<tr>
<td>Low morale due to having “no future”</td>
<td>Low staff morale</td>
</tr>
<tr>
<td>Too high interest rates strangulate any further business</td>
<td>Loss of market share because clients fear business risks</td>
</tr>
<tr>
<td>No trust in leaders</td>
<td>Dilbert cartoons everywhere ...</td>
</tr>
</tbody>
</table>
Principles: Accruing Technical Debt
Practices: 40 Ways to Leave Your Debt
Potentials: Managing the Debt
Not All Debt is Bad – As Long It Is Well Managed

Borrow

Earn money from investments

Repay

Deploy, sell

Invest, cut debts

© 2013 Dr. Christof Ebert, Vector Consulting Services GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector. V 2.0. 2013-01-03.
Case Study: Managing Technical Debt – 1

**Situation:** Market leader in critical infrastructure solutions

- Challenge: Complexity, legacy code, volatile requirements
- Business needs: Flexible solution and service development, schedule adherence, cycle time reduction

**Approach:** Systematically reduce and control technical debt

1) Make technical debt visible
2) Evaluate trade-offs
3) Control technical debt
- **Target**: Focus one team on one business goal

- **Background**: Empowered teams increase productivity – scattered responsibility means no responsibility

- **Approach**: Set up a core team that owns the product. Product manager (business case); Marketing / sales manager (revenues); project manager (project success); Finance / operations manager (allocating appropriate resources)
Practice: Apply Scrum on Product Level

- **Target:** Improve understanding of needs and product implications

- **Background:** Incremental development provides early visibility. Scrum enhances team ownership.

- **Approach:** Scrum approach is extended to the product. Incremental, agile product development with automated change, build, testing
Practice: Focus on Critical Components

- **Target**: Refactor and improve critical components.

- **Background**: 20% of code creates >60% of defects. Critical components accumulate to multidimensional risk clusters.

- **Approach**: Identify critical code based on change and defect history, critical functionalities, complexity, technology, etc. Control refactoring and its reasoning and results – it’s an investment.
Practice: Evaluate Design Decisions and Architecture Impacts

- **Target**: Evaluate design decisions and their impacts

- **Background**: Modeling of design alternatives and architecture impacts shows long-term risks and cost

- **Approach**: Introduce a professional tool chain with workflow support for design and architecture modeling. Ensure a single source for all project and engineering data, such as feature models, requirements, operating scenarios, dependencies, item definition, model-based design
Practice: Apply Static Code Analysis

- **Target**: Focus verification on critical code segments.

- **Background**: Static code analysis helps in identifying defect-prone code segments.

- **Approach**: Run static code analysis (e.g., QAC, Klocwork) based on thresholds such as HIS or MISRA criteria. Relate to architecture concepts such as safety, security. Use dedicated tools depending on the programming language. Create baselines and exception rules to balance the cost-debt ratio.
Practice: Perform Efficient Code Reviews

- **Target**: Apply code reviews where it really matters.

- **Background**: Code reviews have high effectiveness with certain defect types which cannot be efficiently tested.

- **Approach**: Use predefined checklists and dedicated tools (e.g., quickrev) to facilitate efficient code reviews. Connect directly to version control system and focus on the changes.
Case Study: Managing Technical Debt – 3

- Systematically decreased technical debt – where it mattered most
- Efficiency in development and approval, e.g. schedule adherence

![Graph showing delays vs. delivery date with target zone]
Principles: Accruing Technical Debt
Practices: 40 Ways to Leave Your Debt
Potentials: Managing the Debt
Conclusions

- It is vital in today business climate to understand and manage technical debt.

- Three key principles
  1. Make technical debt visible
  2. Evaluate trade-offs and decide
  3. Systematically control technical debt

- Don’t exaggerate the metaphor.
  It’s part of product management, but no pseudo-business.

Our own simple law for managing technical debt: RACE

- Reduce Accidents and Control Essence

Source: Christof Ebert & Reiner Dumke, Software Measurement, 2007
RACE: Reduce Accidents and Control Essence

- Value
- Effort

- Good enough
  - Target
  - Realistic

- Insufficient quality
- Technical Debt
- Overengineering
- Complexity

© 2013 Dr. Christof Ebert, Vector Consulting Services GmbH. All rights reserved. Any distribution or copying is subject to prior written approval by Vector. V 2.0. 2013-01-03.
Industry Challenges to Research and Education

- Is technical debt a **sound metaphor** for managing software and IT investments?

- What techniques help distinguishing **good and bad debts**?

- **Avoid overengineering** and unnecessary investments?

- How to **educate engineers and managers** on technical debt?

- Is technical debt primarily related to **evolution and maintenance**?

- Is there a constructive approach to **budget “repayment”**?

- How to **automatically identify technical debt** in a software or IT project and product?
Global Software and IT
Christof Ebert

Summary of the author’s first-hand experience and expertise, this book offers a proven framework for global software engineering.

"This book stands out as the best source of information on distributed software development. Seldom do we see a book with the concepts completely backed by industry experiences and views. Software developers and managers benefit from the broad case studies."

- S M Balasubramaniyan, Vice President, Wipro Technologies
Software Measurement
Establish, Extract, Evaluate, Execute
Christof Ebert and Reiner Dumke

The book on performance management with lots of benchmark data, case studies and industry experiences.

“Few organizations have really institutionalized measurement of their products and processes. This book is bang up-to-date in both fields and packed with practical advice. For every software engineer.”

- Charles R. Symons, Inventor of Function Points
Thank you for your attention.

For additional information about Vector Consulting Services please have a look at: www.vector.com/consulting

Vector Consulting Services GmbH
Ingersheimer Str. 24
D-70499 Stuttgart
Phone: +49 711 80670-0
Fax: +49 711 80670-444
consulting-info@vector.com

Your Partner in Achieving Engineering Excellence.