

What Makes a Good Software Architect?

2019 Edition

Copyright 2019 Carnegie Mellon University. All Rights Reserved.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

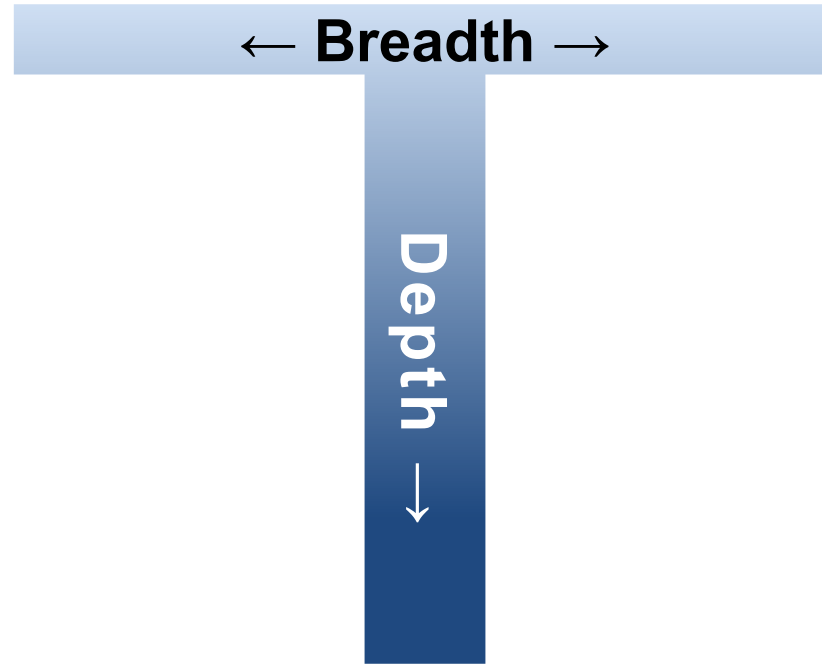
NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

DM19-0347

“T” Model for Skills and Knowledge



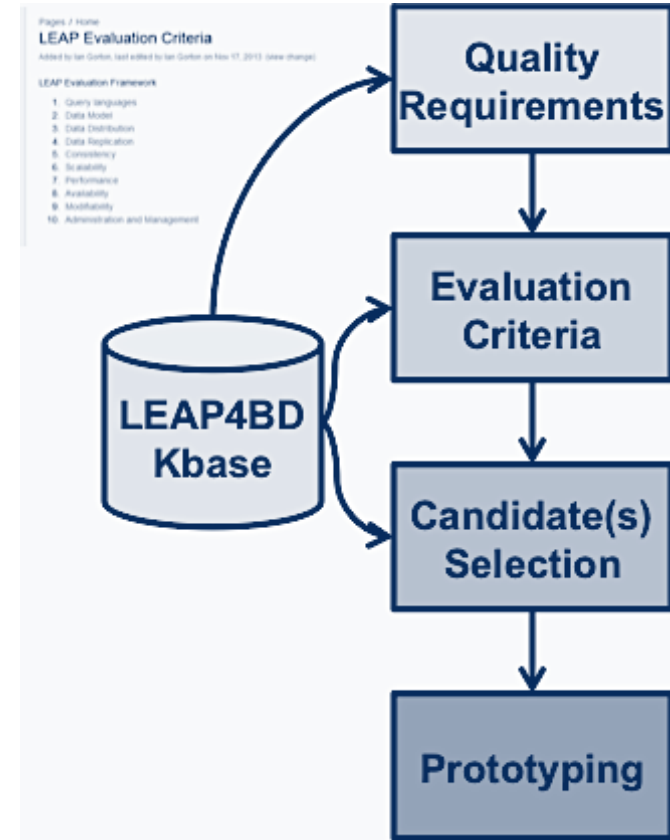
Lightweight Evaluation and Prototyping for Big Data (LEAP4BD)

Aims

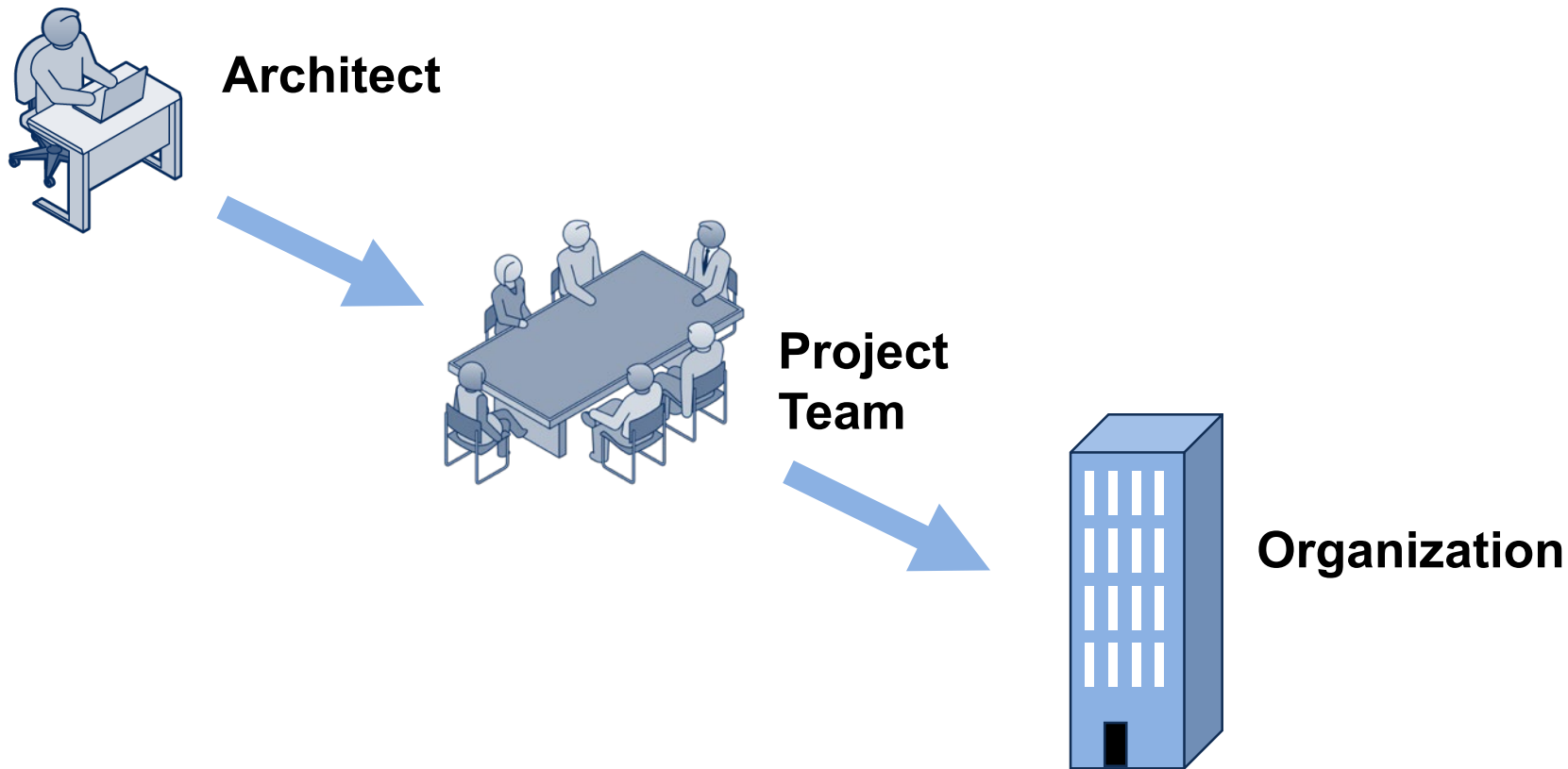
- Risk reduction
- Rapid, streamlined selection/acquisition

Steps

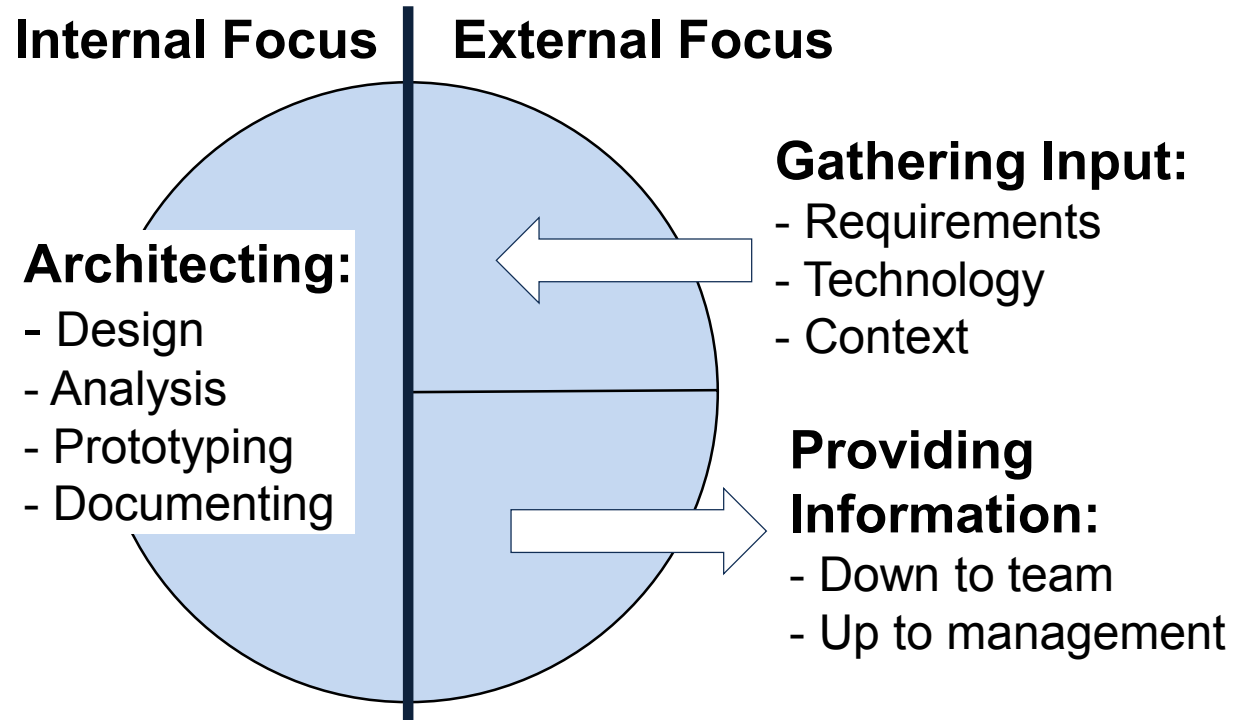
1. Assess the system context and landscape
2. Identify the architecturally-significant requirements and decision criteria
3. Evaluate candidate technologies against quality attribute decision criteria
4. Validate architecture decisions and technology selections through focused prototyping



Architect's Success Depends on Team and Organization



How do successful architects spend their time?



Philippe Kruchten, "What do software architects really do?", *Journal of Systems and Software*, 81:12, 2008, <https://doi.org/10.1016/j.jss.2008.08.025>

SATURN 2019

15th Annual SEI Architecture Technology
User Network Conference

MAY 6-9, 2019 | PITTSBURGH, PA

<http://sei.cmu.edu/saturn>

SATURN is the leading international conference for software architecture practitioners who look beyond the details of today's technologies to the underlying trends, techniques, and principles that underpin lasting success in our fast-moving field.

SATURN attracts attendees from many industrial domains with interest in both research and practice. This unique mix makes it the best place to learn, exchange ideas, and find collaborators at the leading edge of modern software architecture practice.

Join us and become part of the SATURN community!

WHETHER YOU'RE JUST STARTING OUT IN ARCHITECTURE or have many years of experience, SATURN 2019 offers something for everyone. You will leave with dozens of new ideas and solutions to apply in your organization.

The conference fee includes three full days of presentations and events. The SEI also offers one-day courses on microservice architecture, DevOps practices with architecture, and managing technical debt.

Visit us at sei.cmu.edu/saturn

Additional Resources

Technology concerns in big data system architectures

- <https://resources.sei.cmu.edu/library/asset-view.cfm?assetID=90909>

Lightweight Evaluation and Prototyping for Big Data (LEAP4BD)

- <https://resources.sei.cmu.edu/library/asset-view.cfm?assetID=447346>

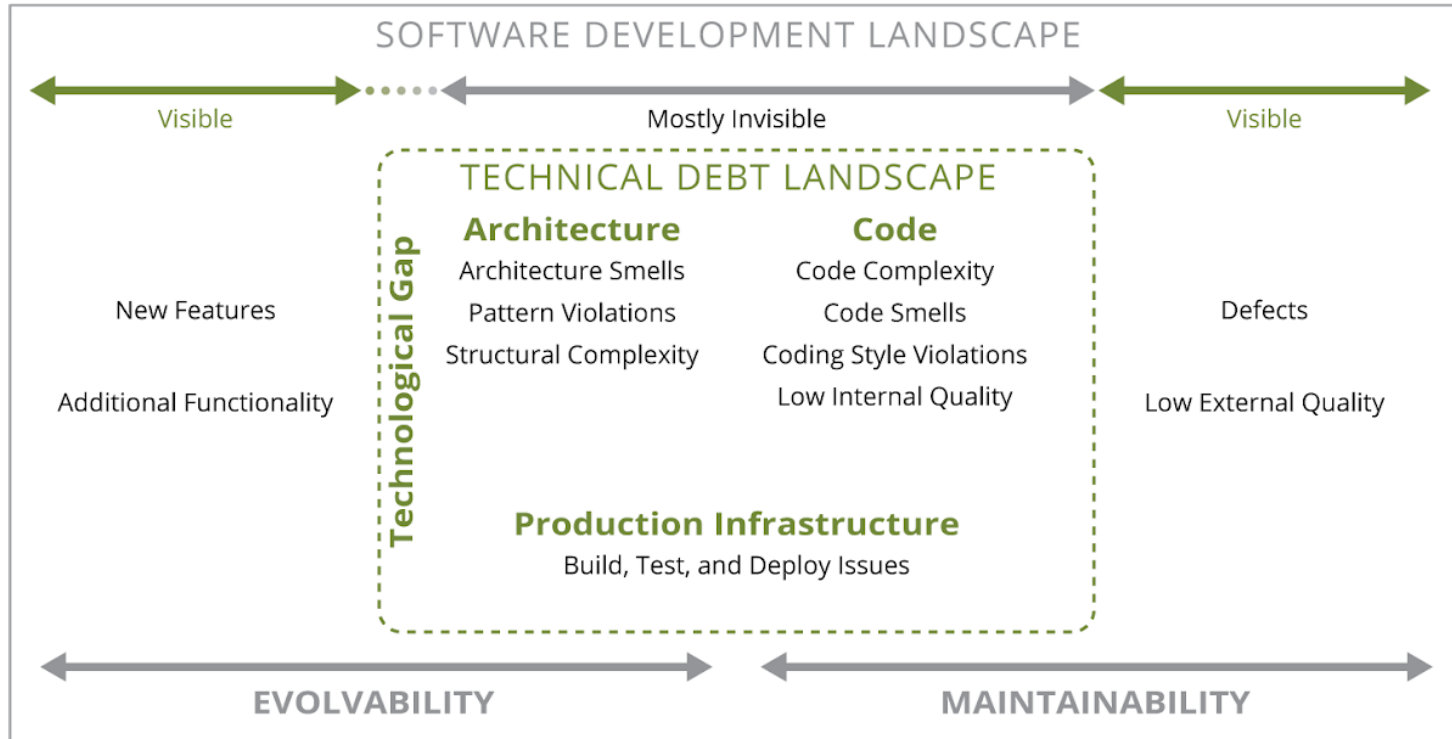
SEI's Architecture Competence Framework

- <https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=509513>

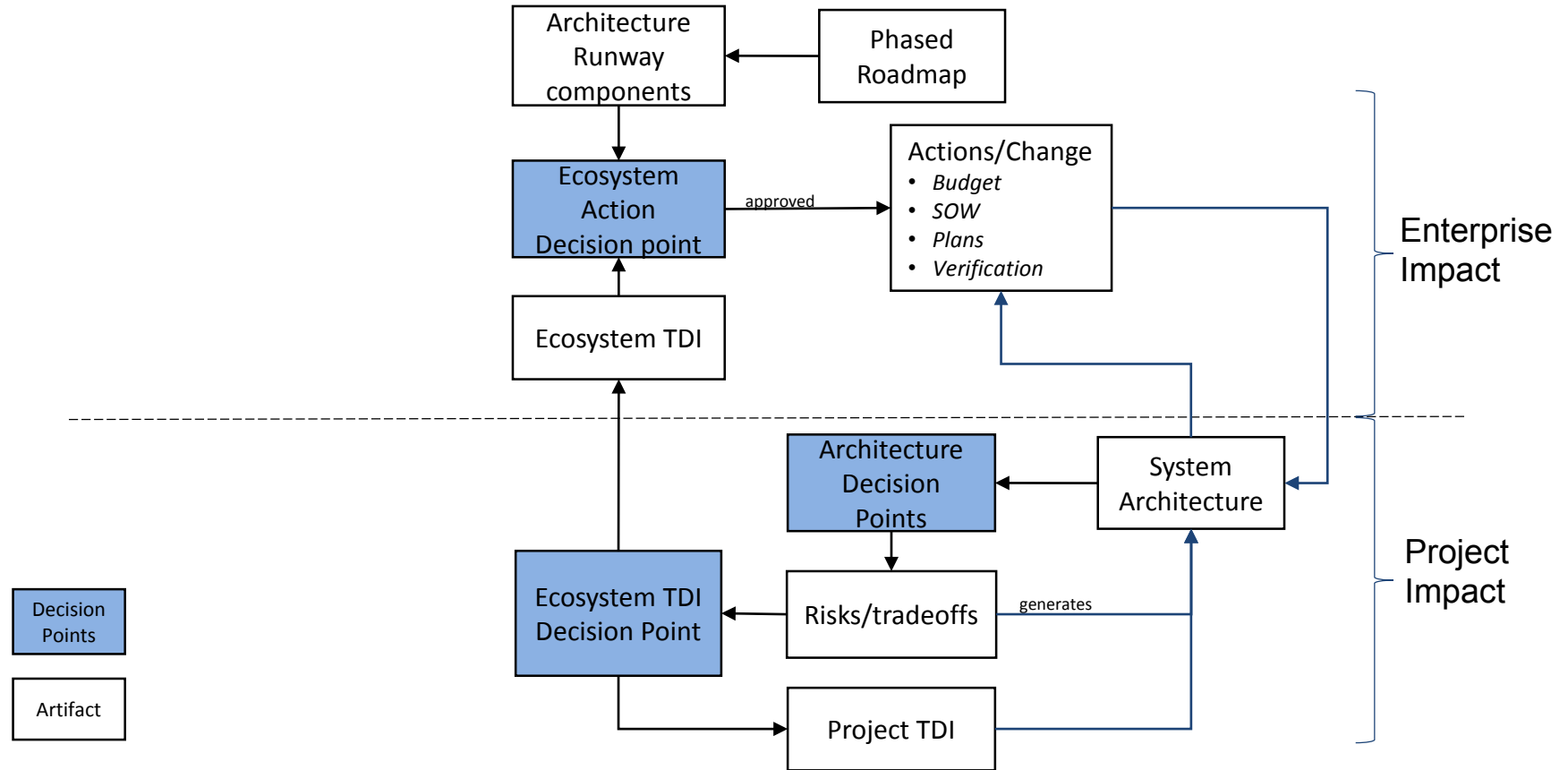
Using the Analytic Hierarchy Process for modernization decisions

- <https://resources.sei.cmu.edu/library/asset-view.cfm?assetID=502340>

Most Technical Debt is Architectural



Architecture Decisions Impact both the Project & the Enterprise



Architectural Technical Debt Examples

- Microservices directive without guidance results in latency and complexity
- Lack of service API for the business apps result in suboptimal and costly solutions
- External service call injecting integer overflow resulting in recurring crashes
- Lack of common integration schema causing rework across projects
- Heterogeneous services causing unintended duplication

Tracking Enterprise Technical Debt

coding convention Code Review

Investigate [] for complex SQL stmts in validation Code Review 1-27 -Application DesignRisk

Reduce branching complexity in WorkUnitDetailMgr.java Code Review 1-27 -Application CodeQuality

Pass the ApplicationContext as parameter to a validator constructor Review Code -API-Application CodeQuality

No common integration schema is used between [] and [] applications IntegrationRedesign options -Integration DesignRisk

[] IS integration mechanism is a mix of ETL/trigger/ database access connecting to physical integration schema IntegrationRedesign options -Integration DesignRisk

Connecting to [] is a system located in the extranet which would require you to cross the intranet/extranet boundary, which has strong security Design Document Review 2-05 - DesignRisk

Grouped by: Labels

Filter Results: TD Indicators

Summary

- Implement service APIs for [] integration
- [] remove duplicative "temporary" fields
- Evolve Reference Data Cache into runway component
- Integrate []
- Revisit [] business rules design for maintainability & modifiability
- Create Organizational Information Shared Service
- New []atures going into old [] application
- Heterogeneous Labs Services
- Microservices directive without guidance may result in latency or complexity issues

1-9 of 9

Two Dimensional Filter Statistics: Open Issues by project

Components	CodeQuality	DesignRisk	TDItem	TODO	T:
Runway	0	1	1		3
-Runway	3	0	0	0	3
ct-Application	0	0	0	1	1
	0	2	0	0	2
.....S Integration	0	2	0	0	2
API-Application	4	1	0	0	5
	0	4	0	2	6
Application	0	3	0	8	11
application	0	3	0	0	3
Application	9	9	0	0	18
ms	0	0	11	0	12
onent	0	0	1	0	2
ique Issues:	16	27	13	12	68

Incorporating Technical Debt into Project Planning

	Visible	Invisible
Positive value	Features	Architecture infrastructure
Negative value	Defects	Technical Debt

Managing Technical Debt: Reducing Friction in Software Development
Philippe Kruchten, Robert Nord, Ipek Ozkaya,
Addison-Wesley Professional, 2019.

Additional Resources

SEI's Architectural Technical Debt library

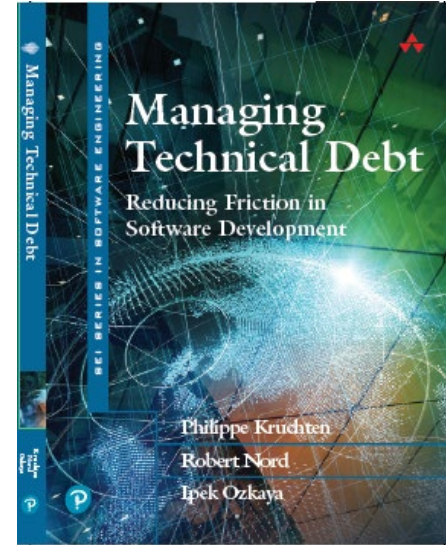
- <https://resources.sei.cmu.edu/library/asset-view.cfm?assetID=509492>

Managing Technical Debt Course offered both online and onsite

- <https://www.sei.cmu.edu/education-outreach/courses/course.cfm?courseCode=P127>

Join researchers, practitioners and tool vendors to share experiences and practices managing technical debt

- <https://techdebtconf.org>



Available May 2019

SATURN 2019

15th Annual SEI Architecture Technology
User Network Conference

MAY 6-9, 2019 | PITTSBURGH, PA

<http://sei.cmu.edu/saturn>

SATURN is the leading international conference for software architecture practitioners who look beyond the details of today's technologies to the underlying trends, techniques, and principles that underpin lasting success in our fast-moving field.

SATURN attracts attendees from many industrial domains with interest in both research and practice. This unique mix makes it the best place to learn, exchange ideas, and find collaborators at the leading edge of modern software architecture practice.

Join us and become part of the SATURN community!

WHETHER YOU'RE JUST STARTING OUT IN ARCHITECTURE or have many years of experience, SATURN 2019 offers something for everyone. You will leave with dozens of new ideas and solutions to apply in your organization.

The conference fee includes three full days of presentations and events. The SEI also offers one-day courses on microservice architecture, DevOps practices with architecture, and managing technical debt.

Visit us at sei.cmu.edu/saturn