Carnegie Mellon University

Copyright 2016 Carnegie Mellon University

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

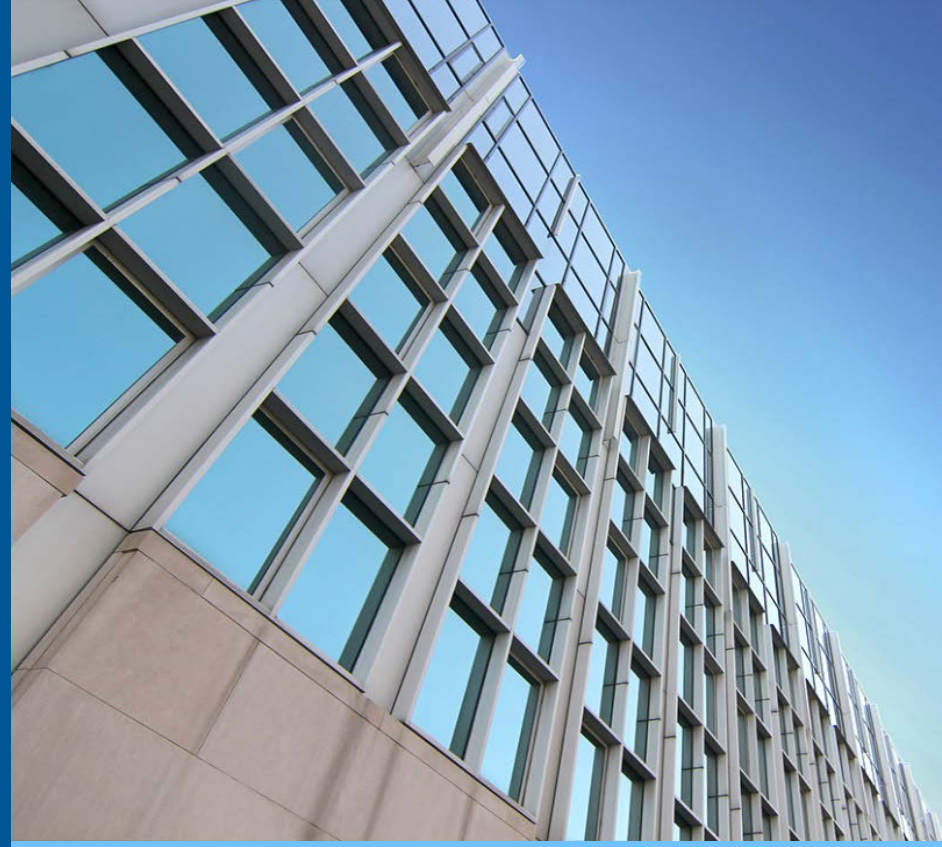
Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

DM-0003417

What Makes a Good Software Architect?

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

John Klein and Andrew Kotov
Hosted by Will Hayes

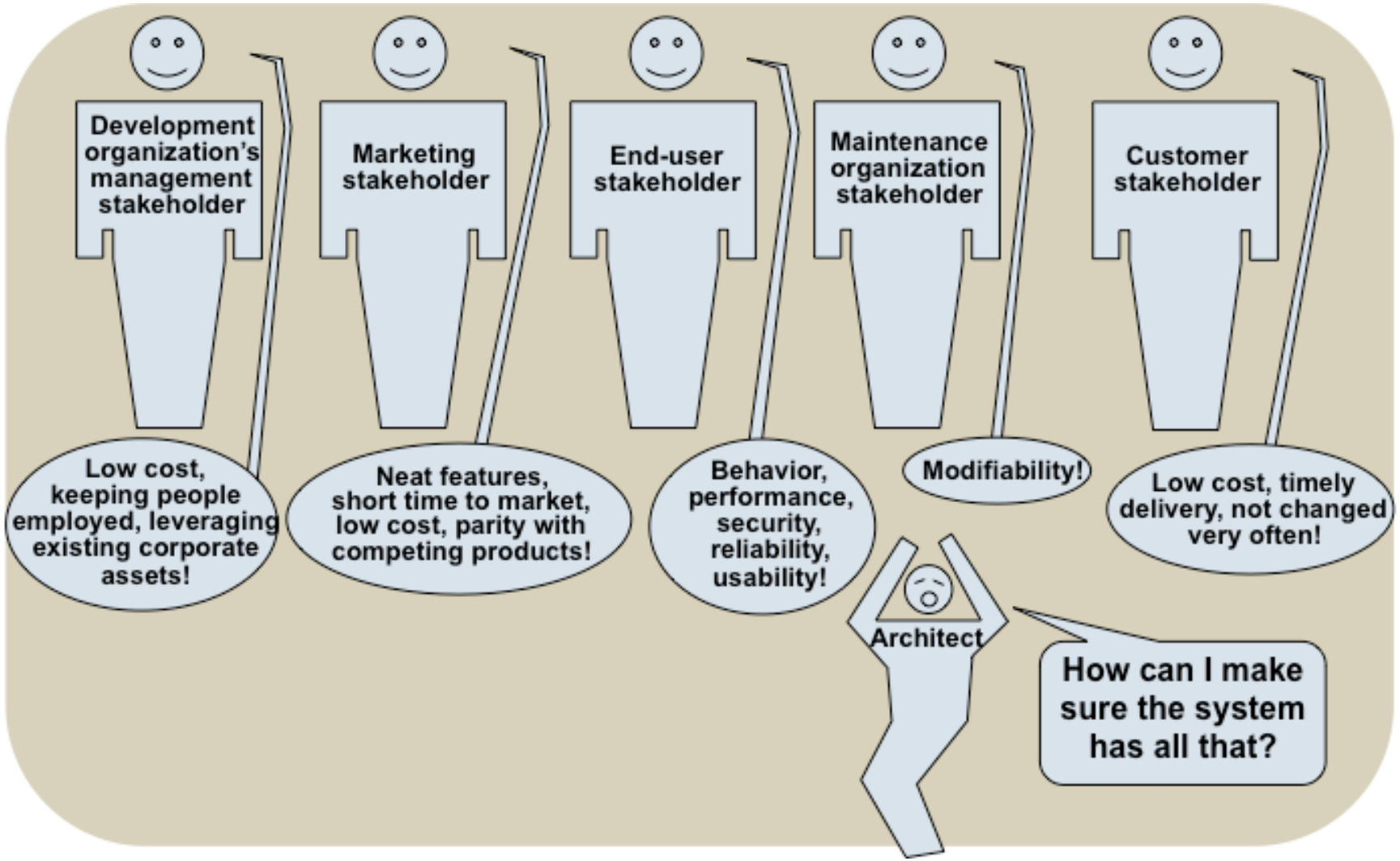


Polling Question

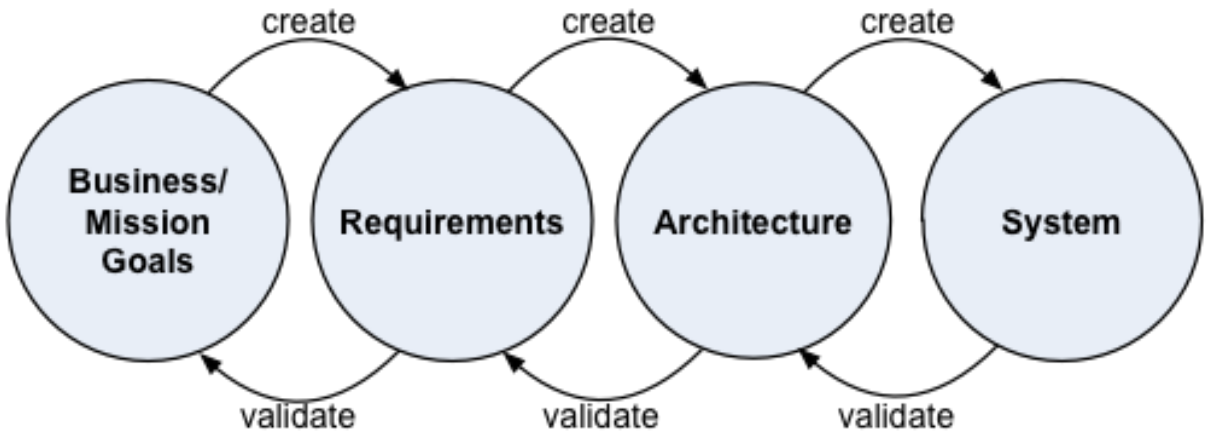
What is your relationship to software architects?

- I am a software architect
- I want to become a software architect
- I manage software architects
- I work on projects with software architects
- Other

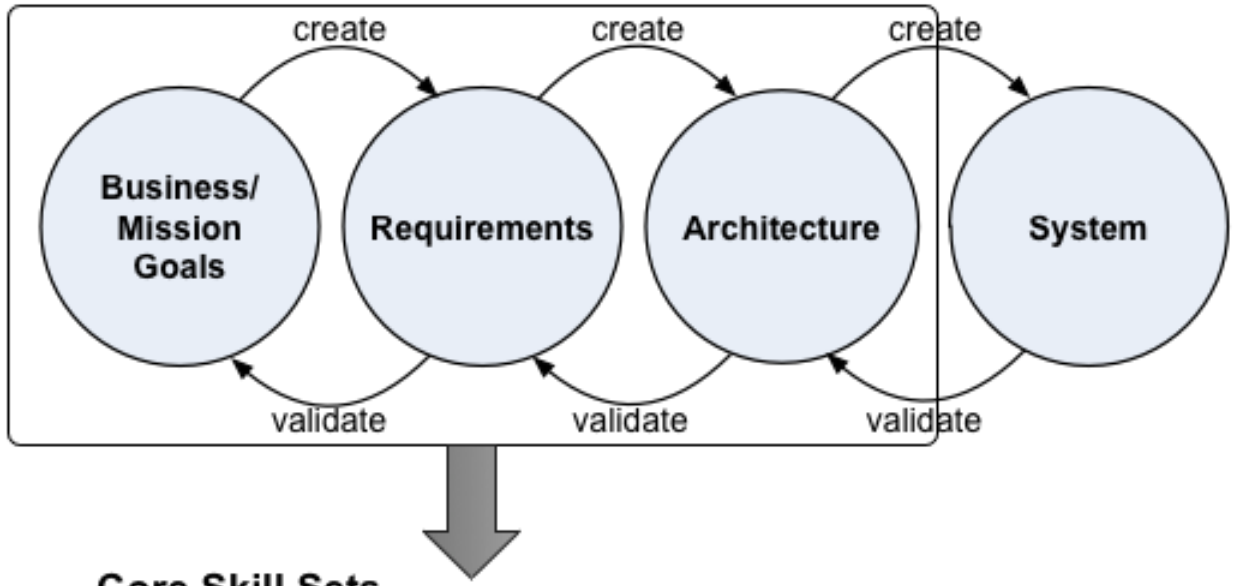
The Life of a Software Architect



What do architects do?



Architect's Skill Sets



Core Skill Sets

- Design - create and evolve
- Analysis - will the design provide the needed functions and qualities?
- Models and representations - "documentation"
- Evaluation - are we satisfying stakeholders?

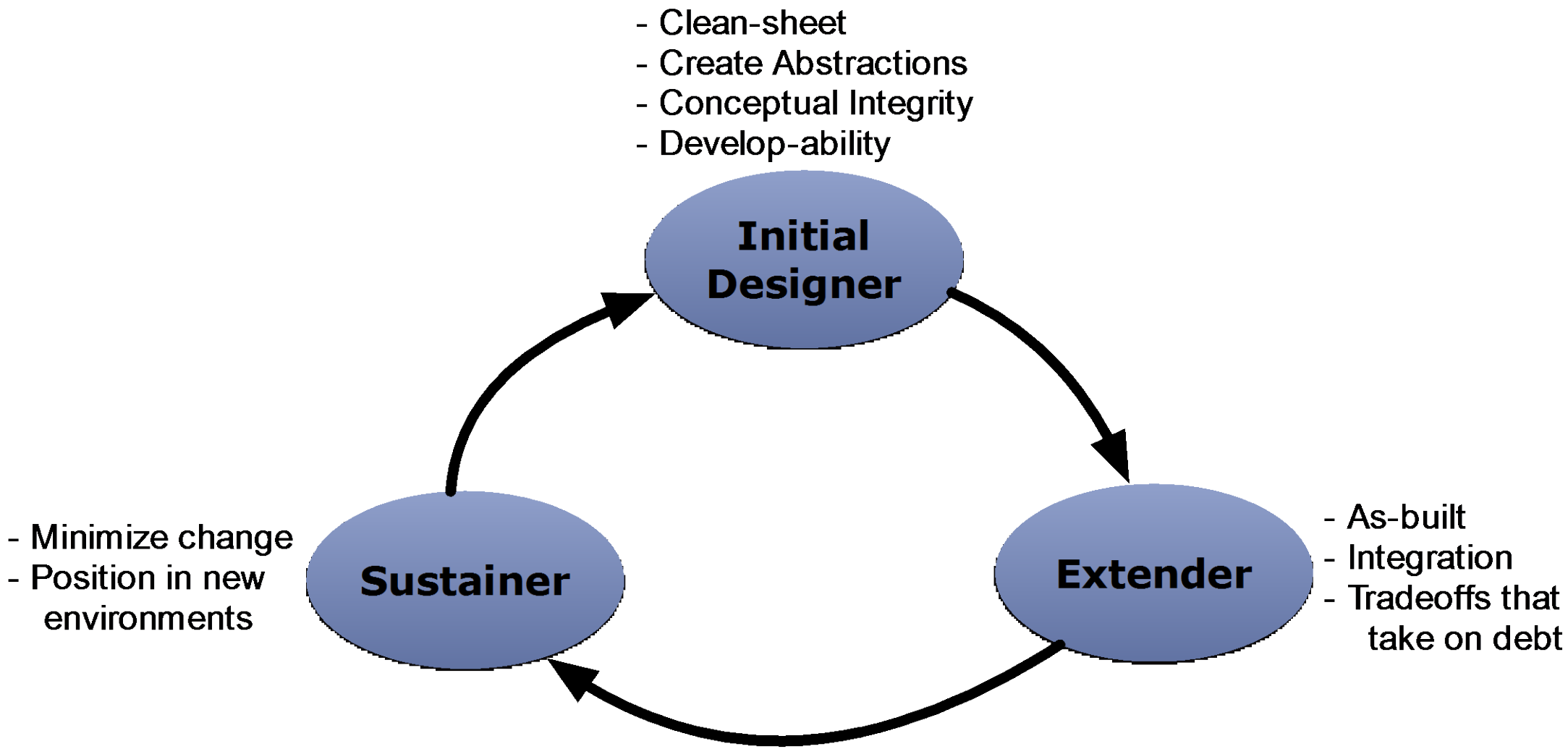
- Communication – with technical and business teams
- Technical Leadership

Polling Question

Do architects in your organization do:

- Architecture design
- Development
- Architecture analysis
- Modeling or other documentation
- Architecture evaluation
- Communicate architecture
- Provide technical leadership
- Provide coaching and mentoring

Architect Skills in the System Lifecycle



Polling Question

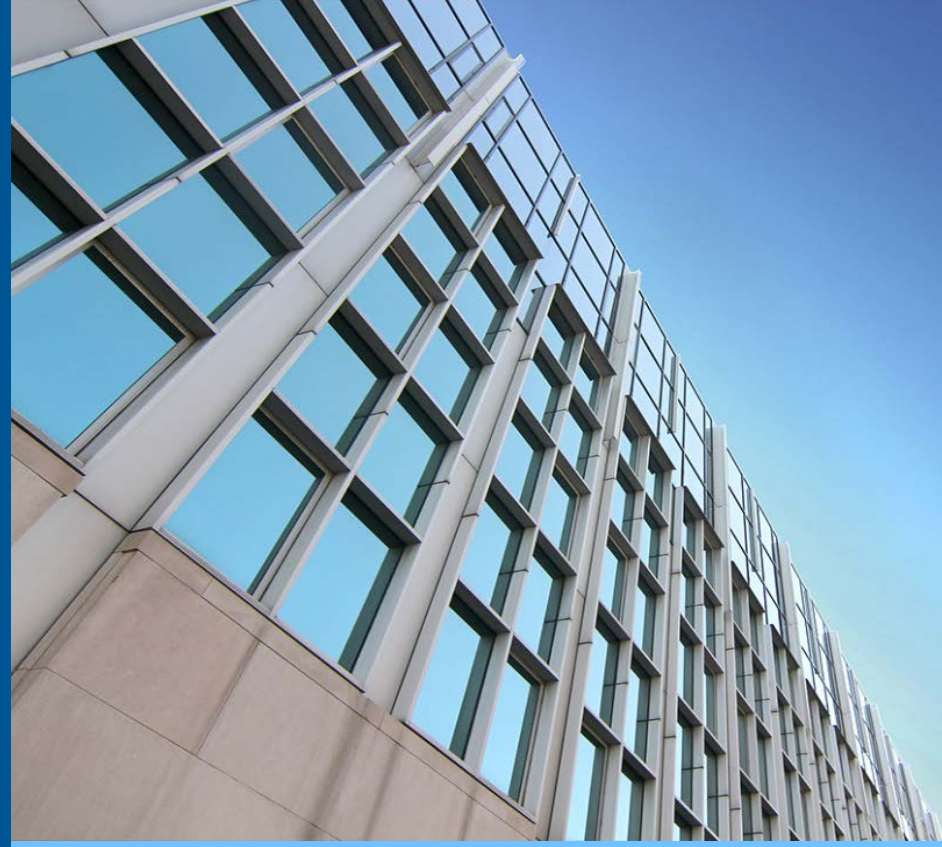
Does your organization offer or require specific professional development for architect (e.g., classes, apprenticeships, certificates)?

- Yes
- No
- Not sure

Architect's Design Trade-off Toolbox: Balancing Agility and Technical Debt

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Ipek Ozkaya, SEI
Michael Keeling, IBM



What is Technical Debt?*

- Exists in an **executable system artifact**, such as code, build scripts, automated test suites;
- Is traced to **several locations** in the system, implying ripple effects of impact of change;
- Has a **quantifiable** effect on system attributes of interest to developers, such as increasing number of defects, negative change in maintainability and code quality indicators are symptoms of technical debt.

* Term first used by Cunningham, W. 1992. *The WyCash Portfolio Management System*. OOPSLA '92 Experience Report. http://c2.com/doc/oopsla92.html.

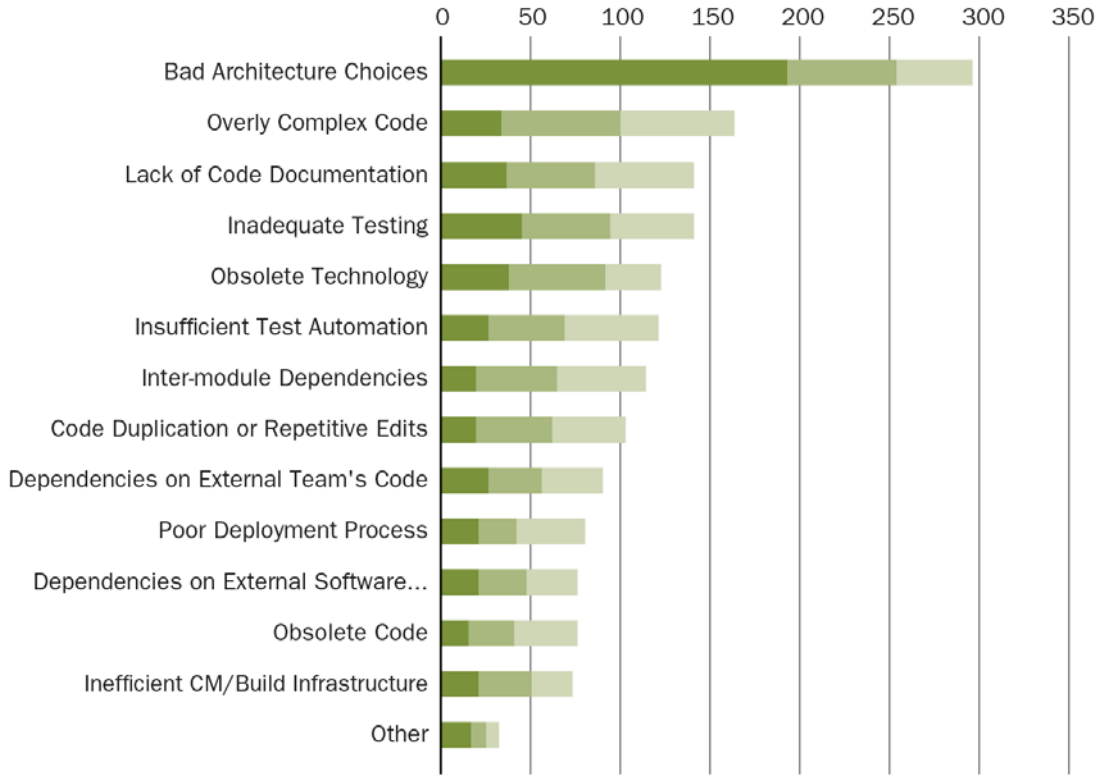
Polling question

Managing technical debt is a critical technical skill that software architects should have.

- I agree
- I disagree

Software Architecture Biggest Contributor

- Bad architectural choices rated as the top contributor to technical debt among over 1800 developers we surveyed.
- 56% of the respondents ranked architecture among top 3 pain points.



A Field Study of Technical Debt **https://insights.sei.cmu.edu/sei_blog/2015/07/a-field-study-of-technical-debt.html**

Polling question

In which of these areas do you observe technical debt the most?

- Code; our code has become very hard to maintain because of clones, cycles, and random bug fixes.
- Architecture; we have made suboptimal architectural decisions that we need to rearchitect soon.
- We have skipped practices such as reviews, necessary testing, and documentation that we are now paying for with low system quality.
- All of the above
- None of the above

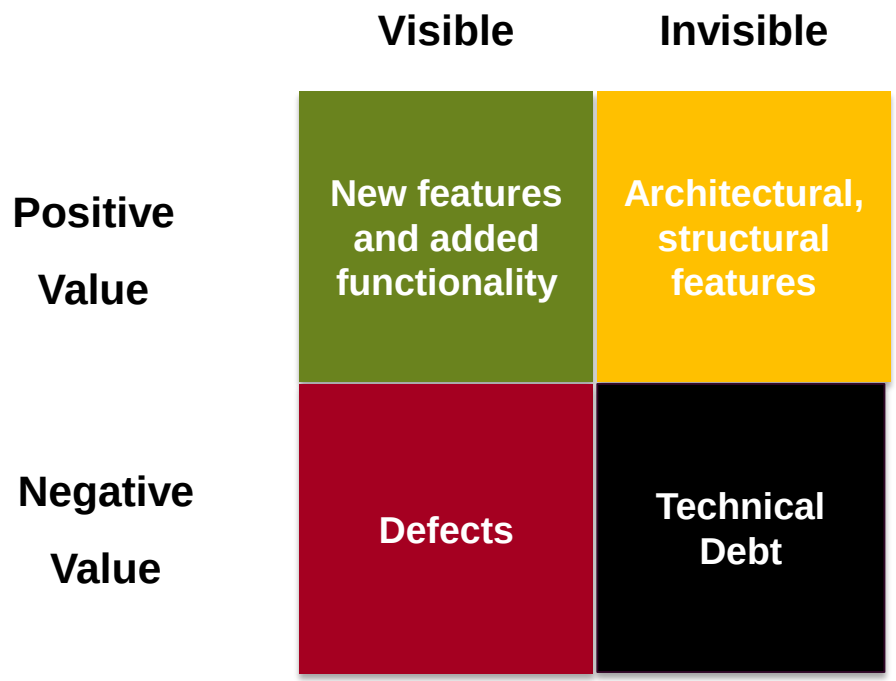
Technical Debt is Not Simply Bad Quality

“we have the source code static analysis tools, but this is to assure proper quality of source code. But how architectural changes are impacting I don’t know.”

Original interpretations of technical debt led us to think it is bad code quality.

- Low internal code quality is a problem, but claiming it as technical debt should not and does not legitimize it!

Essential Software Development Artifacts



Kruchten, P. Nord, R.L., Ozkaya, I. 2012. Technical Debt: From Metaphor to Theory and Practice, IEEE Software, 29(6), Nov/Dec 2012.

Polling question

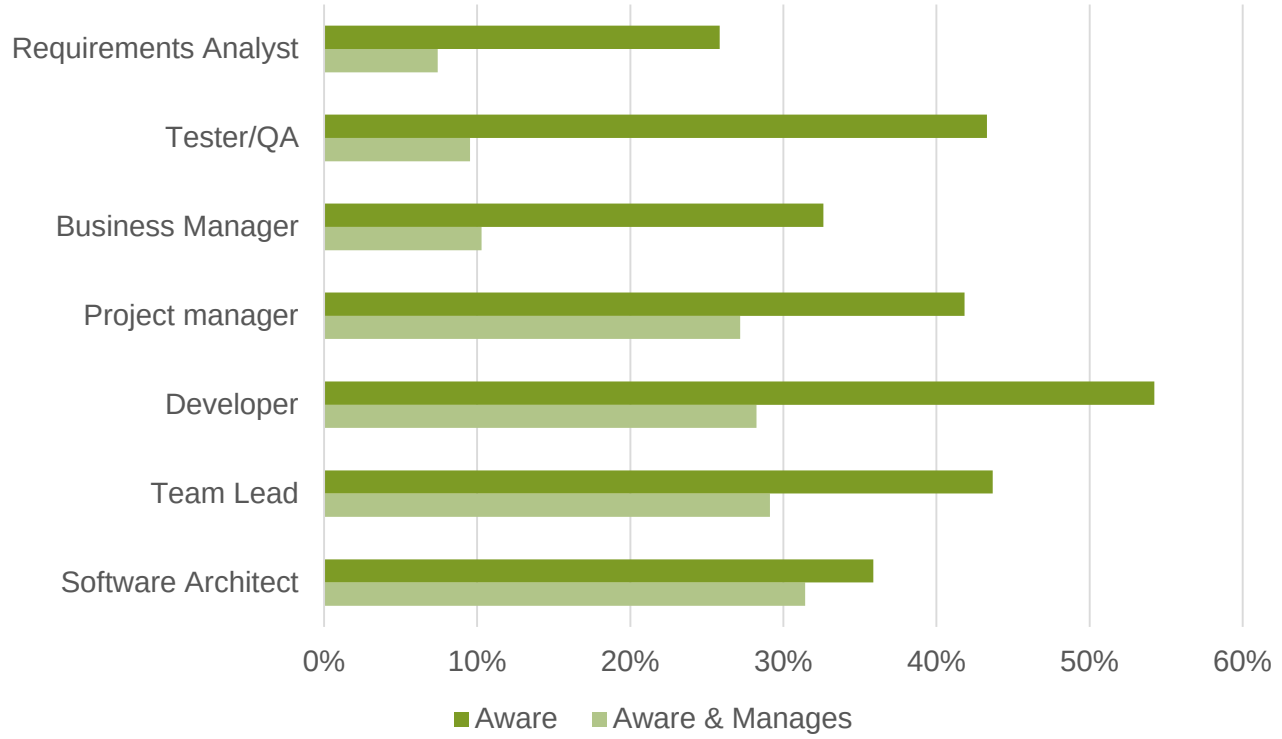
In our project technical debt management is currently owned by:

- The software architect
- The product owner
- The team
- All of the above
- No one

Who is Aware and Manages Technical Debt

Developers are most aware of technical debt.

While a joint responsibility, software architects are reported to own management of technical debt more often than other roles.



A Field Study of Technical Debt **https://insights.sei.cmu.edu/sei_blog/2015/07/a-field-study-of-technical-debt.html**