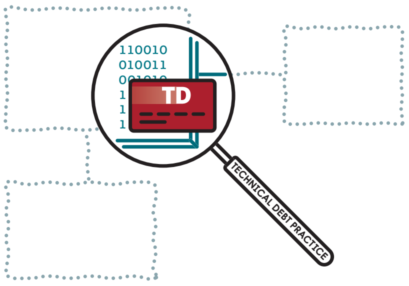
Carnegie Mellon University
Software Engineering Institute

Managing Technical Debt

Identify Technical Debt Items

TECHNICAL DEBT OCCURS when a design or construction approach is taken that is expedient in the short term but increases complexity and cost in the long term. Whether it results from inexperience, accident, or strategy, all software-reliant systems accumulate some technical debt. If managed well, some technical debt can accelerate design exploration and system delivery. Left unrecognized and unmanaged, accumulated technical debt drives up development and sustainment costs. To reason about technical debt, estimate its magnitude, and gain information to make decisions about it, you must be able to anchor technical debt to explicit technical debt items.



Technical Debt Must Trace to the System

Development teams are often aware of the consequences of technical debt, but do not have the mechanisms to communicate the issue. Identifying technical debt items can help. When you trace technical debt to the system, start with the business context, assess artifacts across the technical debt landscape, and record the results as a technical debt item. A description of a technical debt item captures where in the system the debt is located and the associated state of consequences that it causes in the system.

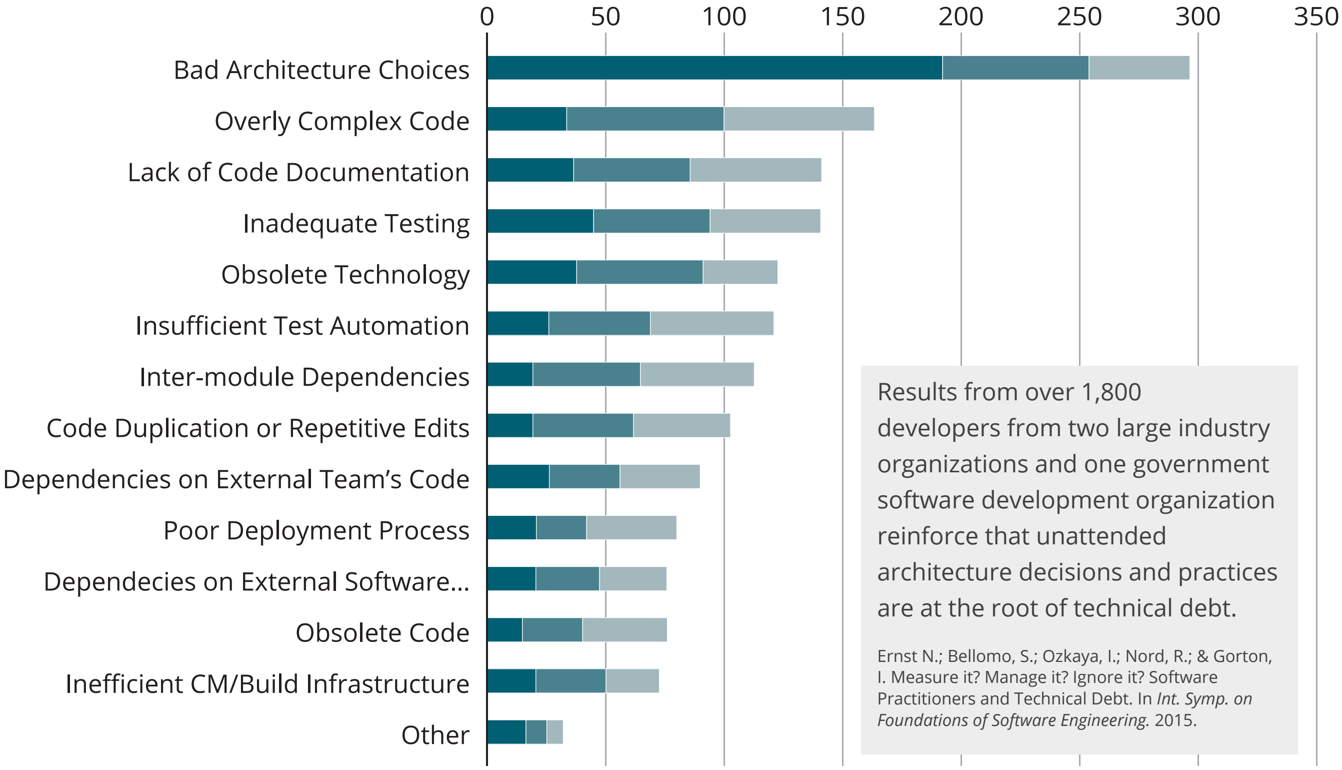
Software developers explain the what and the where as they incur or become aware of technical debt, but they don't highlight clearly the consequences of not fixing it, how the debt might grow over time, and a reasonable time to pay the debt if the fix must be deferred. We recommend building an inventory of technical debt items that record the consequences of accumulating debt. That inventory will help the development team assess how the debt is growing. This practice creates a technical debt awareness mindset for the team, which helps reduce the rate of unintentional technical debt going forward. It also has operational benefits such as the capability to retrieve all outstanding technical debt issues and assess their importance and priority against the team's resources.

Benefits of Identifying Technical Debt Items

Technical debt is a useful concept for fostering dialog between business and technical people in a software development organization. Technical people do not always appreciate the value of shorter time to market and rapid tactical changes of direction. Business people do not always realize the dramatic impact that some earlier design decisions can make in a software project and the cost they can lead to downstream. An organization can help everyone better understand the challenges of software evolution and make the economic consequences more real and tangible by

- identifying concrete items of technical debt
- considering their impact over time
- evaluating lifecycle cost associated with them

Then both technical and business people can plan how to manage technical debt just as they plan new features, fix defects, and evolve the architecture.



Results from over 1,800 developers from two large industry organizations and one government software development organization reinforce that unattended architecture decisions and practices are at the root of technical debt.

Ernst N.; Bellomo, S.; Ozkaya, I.; Nord, R.; & Gorton, I. Measure it? Manage it? Ignore it? Software Practitioners and Technical Debt. In *Int. Symp. on Foundations of Software Engineering.* 2015.

How We Can Help

To meet the challenge of uncovering, communicating, and managing technical debt, the Software Engineering Institute (SEI) has developed a systematic approach to gain control of technical debt. It includes techniques for making technical debt visible and integrating debt management into project planning.

The SEI can help a project team conduct working sessions to identify technical debt items. During the assessment, the SEI will guide the project team—and key stakeholders such as the architect, product owner, and project manager—through activities that build an inventory of technical debt items, which in turn can be assessed for impact and acted on accordingly:

- Quickly identify likely technical debt, and prioritize the items most meriting near-term investigation.
- Collaborate with business and project stakeholders to connect the chosen technical debt items to business goals and the consequence across the organization of carrying that debt.
- Analyze the technical debt items to reveal the concrete software artifact associated with the debt and the rework needed to eliminate the debt.
- Document the technical debt items in the issue tracker.
- Weigh the costs and benefits of the technical debt items to decide which items to address first, which to defer until some measure is reached, or what further analysis is needed.
- Vet the results.

The outcome will be a list of likely technical debt items, analysis and documentation as technical debt items of a select number of the high-priority items, and recommendations that the team can draw on to expand the exercise to the rest of the project. The SEI will review the findings with stakeholders to ensure that analysis is accurate and identify practices they can transition to the rest of the organization.

Contact **info@sei.cmu.edu**.

About the SEI

Always focused on the future, the Software Engineering Institute (SEI) advances software as a strategic advantage for national security. We lead research and direct transition of software engineering, cybersecurity, and artificial intelligence technologies at the intersection of academia, industry, and government. We serve the nation as a federally funded research and development center (FFRDC) sponsored by the U.S. Department of Defense (DoD) and are based at Carnegie Mellon University, a global research university annually rated among the best for its programs in computer science and engineering.

Contact Us

CARNEGIE MELLON UNIVERSITY
SOFTWARE ENGINEERING INSTITUTE
4500 FIFTH AVENUE; PITTSBURGH, PA 15213-2612

sei.cmu.edu
412.268.5800 | 888.201.4479
info@sei.cmu.edu