

Technical Debt is a Systemic Problem - Not a Personal Failing

Technical Debt is a Systemic Problem - Not a Personal Failing

Speaker: Declan Whelan, Christopher Chapman

Duration: 60 min

Track: Process

Level: Intermediate

You often hear technical debt described as a personal failing. Why didn't you code with greater rigor? By creating technical debt, how could you have made life harder on people working in the code? More often than not, technical debt is the result of bigger, systemic problems.

Chances are, you're not a bad person. You didn't want this to happen. It's the system, not you, that's chiefly responsible.

In this talk, we will present some of the conclusions from the Agile Alliance's technical debt working group, which has looked into the systemic causes and consequences of technical debt. While marginal amounts of technical debt will always accrue, that does not explain why substantial technical debt is a widespread phenomenon. The organization in which software development teams work is the much bigger culprit. Many systemic causes, such as deadline pressures, under-investment in skills, and even the unwillingness to measure technical debt, conspire to create a growing burden on software professionals, who would otherwise choose not to create this problem if given the opportunity.

Just as technical debt has systemic causes, the real cost of technical debt lies at the system level. The increasing drag on software innovation has effects not just on individual and team productivity, but on the software value stream, the portfolio, and the organization as a whole. Sometimes, the cost is obvious, such as the valuation of a start-up company's code; other times, the consequences are far more subtle and insidious.

During this session, we will use the language and methods of systems theory to better come to grips with the causes and consequences of technical debt. Don't worry if systems thinking is unfamiliar — we will cover the basics during the talk. We will also do an exercise in which you will create a simple systems model of your own challenges with technical debt, and discuss how this model should help you shape a plan of action for dealing with technical debt.

Ultimately, the goal of this session is to give you the tools to better deal with technical debt. Rather than blaming individual developers, you will be able to show the systemic sources of technical debt, and assess the relative value of addressing each of them. Rather than depending on technical measures to convey the costs of technical debt, we will help you to put the costs of technical debt in stark business terms.

Speakers



Declan Whelan

I help organizations build valuable software. Period.



Chris Chapman

Chris is Owner & President of Toronto-based agile coaching and training consultancy, [derrailleur consulting, inc.](#), Agile Coach partner with [Leanintuit](#).

He has sixteen years' experience as a software developer, team lead, consultant, and coach on many organizations. Since founding [derrailleur consulting](#), his focus has been on how to make good teams and organizations great, and great ones better, through the application of lean, agile, and systems thinking.

Through this, he has worked with organizations from the very small to the very large, across an array of industries and occupations, helping their people uncover ways to understand their organization's systems and how to improve them.

Chris is actively involved in the local and extended agile community, and has spoken at conferences across Canada where he likes to engage participants with live, hands-on demonstrations to help explain the theories of W.E. Deming, Eli Goldratt, and others, and how they can use them in their work.

You can find Chris on [@DerrailleurAgile](#) or local agile events around GTA.