ARLO BELSHEE

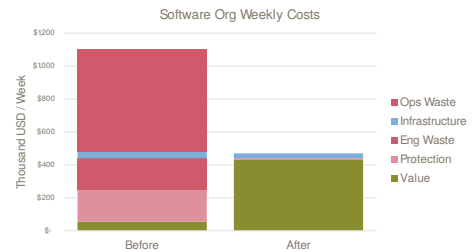*PRINCIPAL SOFTWARE ENGINEER AND MENTOR*

arlo@belshee.com
(503) - 875-1234
Twitter: @arlobelshee
Linked In: arlobelshee

Profile

I speed up my teams. We write fewer bugs and take less time to get stories done. This happens because I know how to reduce technical debt in ways that have immediate positive return during the current story. I both do these myself and teach others.

Software Org Weekly Costs

Selected publications

How I Design Systems
The No Mocks Book (2012), and Architecture — Scaling Design (2013)

I build systems with less coupling than many people. The key aspects are higher coupling and conceptual integrity, architecting the people as part of the system, along with use of more testable architectural patterns. These articles provide examples.

http://arlobelshee.com/the-no-mocks-book/

http://arlobelshee.com/architecture-scaling-design/

How I Prevent Bugs
I Find Bugs Too Boring to Write (2017)

My teams get to Zero Bugs, whether I'm their manager or a member-leader. This talk discusses my reproducible approach.

https://www.deconstructconf.com/2017/arlo-belshee-i-find-bugs-too-boring-to-write

How I Help Teams Improve Their Practices
Agile Engineering Fluency (2016)

The hardest part of change is helping people see what they don't know and have never experienced. Here's how I solve that for engineering improvements.

https://www.youtube.com/watch?v=o6rQNwYLy6A

Skills

Mend Legacy Code
Safely modify unfamiliar code quickly. Address systemic flaws as a series of small, safe changes. Mentor others to do same.

Agile Leadership
Lead teams' ability to responsibly own their outcomes. Mentor individuals and groups towards increased autonomy and inclusiveness.

System Architecture
Smoothly transition large systems between architectures. Guide teams on architectural choices and paths to get there.

Technical

JavaScript / C# / C++ / Python	XP / Lean / Kanban	Microservices / REST APIs
OData / REST	People Management	DevOps Enablement
SQL	Emotion Dynamics	Incremental Rewrites
TDD / Refactoring / Paring	Vertical Leadership Development	Hexagonal Architecture

Selected Experience

Tableau
Principal Agile Development Coach
2016-2019

- Designed solutions to incrementally shift C++ monolith code to new microservices architecture.
- Created formal training program to teach 600 developers to safely refactor untestable legacy code to the

point they could TDD it.

- Created and guided 13-division program to eliminate technical waste for 1000-person technical team.
- Mentored cross-division architecture teams about architectures that support teams shipping separately.
- Paired, mobbed, and/or whiteboarded with teams to help them solve design problems.
- Advised CTO and directors on management structures and processes to implement Agile transformation.
- Mentored managers on how to lead Agile teams.
- Mentored Scrum Masters on how to integrate agile techncial and team practices.

Results

- Teams I worked with reduced story cost by 40% and bugs written by 50%, while other teams in same organization exhibited no significant change.
- Leaders I mentored were able to delegate significant new responsibilities to their teams, decreasing delay time for decision-making from 1 week to 1 hour.

Testimonials

> No other person in the world has shifted my perspective on software development, and software development organizations, as much as Arlo. He lives inside a worldview of the future that most haven't glimpsed, some haven't grasped, and even fewer see how to get there.
>
> — James Baker, Principal Technical Advisor, Tableau Software

> I just attended the Read by Refactoring workshop from @arlobelshee and it was awesome! He's one of the top minds in our industry!
>
> — David Bernstein, Author, Beyond Legacy Code

Microsoft   2011-2014
Senior Software Developer, Engineering Excellence

- Paired, mobbed, and/or whiteboarded with teams to help them solve design problems.
- Dropped in to code with teams (in mobs) across the company to address whatever design flaw was blocking them and get their code testable, regardless of their language, tools, or codebase.
- Created formal training program to teach MS Office teams to safely refactor untestable legacy code.
- Advised tools teams on how to shift towards DevOps on a 35-year-old codebase.
- Identified teams across the company that needed help with technical waste, found root causes, aligned their management and individual contributors, and then delivered that help — whatever it was.

Results

- Reduced bugs written in a large-scale MS Word project from apx 5000 in prior release to 3 in the release I helped with, while shortening project duration from 2.5 years to 4 months.
- I worked with half the teams creating a new product. My teams implemented 2/3 of the featueres, in only 80k lines of code, with about 1 bug per team-week. The other teams implemented 1/3 of the features, taking 1.5M lines of code and about 100 bugs per team-week. My teams finished early and then helped recover the other teams.

Testimonial

> Arlo was a key partner in helping the Microsoft Office organization transition to an agile continuous delivery model, with a focus on refactoring legacy code and getting it under test.
>
> I sat in on a lot of his "mob sessions", where we'd bring in a piece of code and have Arlo coach us on how to get it under test. My favorite saying from Arlo was "your bug database should be the corner of your whiteboard...and it should be empty" :)
>
> — Dheeraj Sarpangal, Principal Group Engineering Manager, Microsoft

Industrial Logic   2009-2011, 2015-2016
Principal Software Developer, Principal Technical Consultant

- Extended an IDE plugin and the service that backs it.
- Software archeology to bring in-house a codebase that had been outsourced for a decade. Made it testable and build internal knowledge.
- Trained client companies in saving legacy code, TDD, refactoring.

Results

- Only 2 bugs reported in my IDE plugin over 5 years starting with first delivery.
- Other individuals took over the IDE plugin after I left; design was intact when I returned 5 years later.
- New Read by Refactoring training course brought in about $300k in revenue while saving money for our clients.

Testimonials

> #ReadByRefactoring is the most pragmatic solution for tech debt that I've ever seen. Thanks @arlobelshee!
>
> — Aaron Silinskas, Principal Software Engineer, Expedia (client)

Silver Platter Software

1999-2003

Architect & CTO

- Assembled a start-up technical team and grow it from 3 people to 50 people over several years.
- Lead transformation to Extreme Programming.
- Architected core solution for 2 products.
- Provided code-security guidance to organization.
- Coded large portions of our initial product.

Results

- 2 products brought to market. First grew to $500k annual revenue. Company folded for investment reasons a couple months after beginning sales for second.
- After folding, each team member in the technical org went on directly to their dream job. Some became VPs, others Sr engineers at Google and similar companies.
- Invented several processes that influenced the early Agile community, particularly Promiscuous Pairing.

Testimonials

> Arlo's facility with programming language concepts and architectural design is superlative and working with him every day quickly inspired me to become an expert and approach programming with rigor and precision.
>
> Six years later, I still miss programming with Arlo.
>
> — Jon Stewart, Vice President, Solutions Development, Stroz Friedberg

---

Education

Pomona College - Claremont, California

Math major

Arlo Belshee — arlo@belshee.com — (503) - 875-1234