Donna Malayeri

@lindydonna | www.cs.cmu.edu/~donna | lindydonna (at) gmail.com

Objective

Setting direction and strategy for high-impact, technically-intriguing software projects.

Areas of Expertise

Languages: F#, C#, Scala, Java

Design of developer tools and technologies

User interface design

Education

Carnegie Mellon University (Pittsburgh, PA)
PhD computer science, 2009
Thesis topic: language support for post-hoc object-oriented reuse (Advisor: Jonathan Aldrich)

MS computer science, 2005

University of Maryland (College Park, MD)
BS computer science, 2001

Experience

Microsoft (Redmond, WA)

Program Manager *Nov 2013–Present*

- Program manager on Azure Mobile Services, a hosted service tailored to mobile apps
- Driving technical design of the offline sync feature; creating collateral such as tutorials, samples, and technical presentations
- Ensuring a great developer experience by gathering input from beta customers, coordinating frequent releases from the dev team, and ensuring feature discoverability
- Developing sales and marketing materials for use by the field; engaging with developer evangelists; creating content for hackathons and workshops.

Software Engineer *Feb–Nov 2013*

- Project manager and software developer on Reactive Extensions (Rx), a library for composing asynchronous and event-based programs
- Simplified and streamlined build system in order to quickly incorporate community contributions
- Evangelized use of Rx among product groups within Microsoft; drove the integration of Rx into public tools and libraries
- Engaged with external community and presented popular talks at developer conferences

Program Manager *Aug 2011–Feb 2013*

- Served as project and product manager for F#, a functional programming language on Microsoft's .NET platform
- Drove technical direction and strategy; ensured day-to-day alignment of test and development team; managed schedules and deliverables; scoped product features; coordinated with stakeholders in related product teams.
- Engaged with customers through a variety of channels; used customer feedback to inform improvements in usability, discoverability of features, and product documentation.
- Evangelized F# both internally and externally; presented several well-received talks at key industry conferences.

Experience

Scala Team, EPFL (Lausanne, Switzerland)

Postdoctoral Researcher *Oct 2009–Jul 2011*

- Worked on research team that developed Scala, a programming language with full Java interoperability and a strong type system.
- Improved stability and performance of the Scala plugin for Eclipse, using product feedback from customers and internal users.
- Redesigned and implemented Scala REPL (read-evaluate-print-loop) to tightly integrate with other IDE features in Eclipse.
- Performed formal usability study on Scaladoc; improved layout and design to improve clarity; showed that study results can be applied to similar languages.

Google (Kirkland, WA)

Software Engineering Intern *Jun 2007–Aug 2007*

- Designed and implemented a full-fledged Eclipse debugger front- and backend for the Rhino JavaScript-to-Java compiler, to allow debugging of server-side JavaScript.
- Instrumented existing programming environment to support the new language syntax; added support for auto-completion and other advanced IDE features while editing server-side code.

Microsoft Research (Redmond, WA)

Research Intern *Jun 2003–Aug 2003*

- Developer on the Fugue software protocol checker, a tool for ensuring that programs correctly maintain object state invariants.
- Designed annotations for specifying object invariants on exceptional control flow paths; implemented new analyses for assuring correctness of exception-handling code.

TRW Inc. (Reston, VA)

Software Engineer *Aug 2001–Aug 2002*

- Worked on design and implementation of large C++ project for battlefield simulation
- Developed, tested, and performed on-site deployment of the software.

Research

Carnegie Mellon University (Pittsburgh, PA)

Graduate assistant, Computer Science Department *2002–2009*

- Published 5 peer-reviewed conference articles.
- Designed a new, statically-typed language to allow safely adapting code without modifying it directly. Language includes features which previously required dynamic typechecking.
- Performed empirical study of existing Java programs, which showed that the new language could eliminate many runtime checks and would make programs easier to maintain.
- Designed and implemented a Java extension with a novel multiple inheritance mechanism, to help eliminate code duplication and promote reuse of library code.
- Redesigned Java exception specifications to reduce annotation overhead. Implemented a language extension and tool for specifying and enforcing exception policies.

Additional

Strong communication and writing skills

US Citizen