ERIC RICHARDS

Principal Software Engineer Lead, HoloLens

Microsoft Corporation
1 Microsoft Way
Redmond, WA 98052

www.linkedin.com/in/ericri
www.ericri.com
Eric_Richards (at) ericri.com

 E**RIC**R**I**

Summary

I enjoy being a key member of an exciting technology-driven product team and working with passionate and enthused peers. I lead shipping great quality software, leveraging my deep technical expertise and management experience as a development lead in Windows Holographic / HoloLens, Windows 8 Inbox Windows Store Apps, in Windows Live Photo Gallery and a lead-of-leads in Microsoft Office.

- I have years of commercial software development experience, seventeen of those at Microsoft. I've worked on products from V1 through multiple releases for consumers, information workers, scientific researchers, and industrial technicians. My core focus is on user experience with a background in understanding user psychology + physiology.
- Skills I've developed include excellence at technical management, cross-group technology adoption, dealing with and leading through the ambiguity of a new product, building my team members, adopting and integrating complex technology into a large C++ code-base, and customer focus + engagement.
- At Microsoft, I've shipped the Microsoft Windows 8 People App, features in the Mail and Calendar apps, Photos App, Windows Live Photo Gallery Waves 4 / 3 / 2 and InfoPath 2007 / 2003 SP1 / 2003. I am exceptionally familiar with the rhythm of the product ship cycle for both consumer and business client software.
- Courses of note: Adaptive Leadership (VP nominated class), Scaling Up, Situational Leadership II, Career Coaching, Crucial Conversations. I also participate as a mentor in the Microsoft mentoring program.
- I've been developing software recreationally since 1981 (first computer - a TRS-80 Color Computer). I have deep experience with C and C++.

Experience and Expertise

Results

I ship great software. I ship features that are on-time, high-quality, and at a known-state of development during the entire process of design, implementation, and stabilization (and then further into sustainability). I ensure my team members grow to appreciate what it means to implement user-focused client software.

Management

I've led teams ranging from 5 developers to 12 developers (with two direct dev lead reports) on teams of 30 or more total developers. All of my teams have had strong cross-team deliverables and dependencies to successfully manage.

Technical Knowledge

Core: C++, C++11, C++14, C++/Cx, C, JavaScript / ECMAScript5

Other: VBScript, C#, x86, Perl, Pascal, Prolog, LISP, PL/I

Technologies: HoloLens, WinRT, WWA, COM / IDL, Trident (IE), DHTML, XHTML, HTML, HTML5, CSS, CSS3, XML, XSLT, XPath, Win32, AJAX / XMLHttpRequest, Web Services (SOAP / REST), WebDAV, windbg, STL, Visual Studio, Lex, Yacc, rpcgen.

OS: Windows 10, Windows 8, Win7, Vista, W2K8, W2K3, XP, SunOS, Unix, QNX RTOS, Microware OS9 RTOS, Apple OS.

Product Group History

Windows 10 - Windows Holographic / HoloLens – Development Lead
2013 – *Present*

Just announced at the beginning of 2015, HoloLens is an augmented reality experience for Windows. While HoloLens for Windows 10 is announced, details are still limited. I am a Software Engineer Lead for an agile, adaptive team delivering a high-profile end-point of the HoloLens experience, ensuring the grand orchestration of technology works just right.

Windows 8.1 Store Mail and Calendar Applications – Development Lead
2012 – 2013

My team delivered key new features for the Mail and Calendar applications, including a new organizational view of Mail (the Power Pane, with a People-centric view) and a high-level view of your Calendar (Agenda view).

Growth: tight cross-group coordination of a new, complicated feature on an aggressive schedule.

Delivered: high quality, resilient implementations within an aggressive time frame; prepped the team for transition to completely new org.

Windows 8 Store People Application – Development Lead
2012

I switched to be the Development Lead for the Windows Store People Application to drive a larger application effort key to Microsoft's default service-connected address book and social network experience for Windows 8.

Growth: I ramped up quickly on a new deeply JavaScript application with complicated, high-performance custom controls for virtualized data and ensured it shipped on-time at high-quality. I also ramped up on the variety of connected services the app depended upon, like ABCH, SUP, WNS, and the Facebook Graph and REST APIs.

Delivered: final People Application inbox with Windows 8 plus profile picture editing feature from our summer intern.

Windows 8 Store Photos Application – Development Lead
2010 – 2012

I was a co-lead to a large effort to write from scratch a new Photos app to be delivered inbox with Windows 8. In addition to working with local files, it connected to services like Facebook photos, flickr, SkyDrive, and photos on remote computers. We choose to write a WWA and use HTML5, CSS3, and JavaScript as the main view along with C++ for core logic and platform.

Growth: The biggest challenge here was leading development on a brand new app store product based on HTML5 + JavaScript + C++ on an aggressive schedule with a constantly growing and changing platform (Windows 8 WinRT). Much cross-group work was required in addition to being adaptive to breaking changes.

Delivered: my team designed and implemented the Photos App 1-up viewer, printing, device play-to, photo import experience, selection, and hostable aspects of the photo viewer framework. Additionally, lead an intern effort to integrate Microsoft Research's AutoCollage into the legacy Photo Gallery application.

Windows Live Photo Gallery Wave 4 – Development Lead
2009 – 2010

I led one of the dev teams that delivered the huge (and very well received) Wave 4 release of Photo Gallery. My team's features involved a major reworking of the interface to support the Ribbon command UI and adding face recognition and batch people tagging to the product (cross-group collaboration with Microsoft Research in

Redmond and China). Additionally, group shot technology from MSR was integrated as part of the Photo Fuse feature (choosing the best features of multiple photos to create one composite). Underlying architecture of the photo graphics engine and some of the display technology was upgraded.

Growth: Exceptionally tight schedules for aggressive and large multi-milestone features required a large amount of craftsmanship to complete at high-quality. Multiple cross-group dependencies greatly increased the challenge of achieving these results. Educated new team members on what it takes to be a great photo organizing and editing environment and brainstormed on key feature flow, like batch face tagging.

Delivered: integration with Microsoft Research technology for Photo Fuse (user driven combination of multiple pictures into one) and Face Recognition and Grouping; an end-to-end story for face recognition and batch people tagging to complete the People features in Photo Gallery; integration of the Ribbon command technology; batch photo edit; and updated photo graphics engine and new list-view display technology.

Windows Live Photo Gallery Wave 3 – Development Lead
2007 – 2009

My team added the integration of people tagging to Photo Gallery, driven by face detection technology from Microsoft Research to aid in clicking on a detected face and assigning a name.

Growth: in addition to managing difficult cross-team feature work, I took on additional reports and areas of responsibility when a development lead peer left and his areas of ownership (edit and sync) transferred to me. Also, I helped the product team and the Live Division to adopt the Feature Crew implementation process to ensure quality upfront in feature development.

Delivered: My team shipped People in Photo Gallery (including face detection and address-book aware people tagging for photographs), integration of photo noise reduction, integration with Live Writer and installable extras.

Windows Live Photo Gallery Wave 2 – Development Lead
2007

I joined the Find and Share team of Photo Gallery in the middle of Wave 2. My first responsibility was to coordinate what it would take to successfully ship an Alpha of Photo Gallery, ensuring that feature work was balanced to have a useful Alpha experience and a path to shipping Photo Gallery for Wave 2.

Growth: coming up to speed quickly on a consumer product in a media (photo and video) space in order to make good judgment calls about what work to do when and which issues needed to be fixed during the various milestones of the release. Also, helping reports make proper career choices (inside/outside of Microsoft). I advocated for adoption of the Feature Crew development process, trying it out on two late features so that leadership could see the benefit of using Feature Crews to create high-quality, stable features.

Delivered: flickr photo publishing integration, frame window UI, photo publishing UI, and setup for Photo Gallery.

Microsoft Office InfoPath 2007 – Principal Development Lead / Lead-of-leads
2004 – 2007

I managed a crack team of developers that delivered InfoPath technology integrated into Outlook, Word, Excel, and PowerPoint, along with providing a hostable InfoPath editor control and integrating the InfoPath designer with Visual Studio technologies. I was also InfoPath's triage representative, from the B2TR release through RTM, ensuring that the team kept a high-quality bar for picking which issues to fix and then making sure these issues made it through Office Triage and didn't cause regressions.

Growth: as a manager and a contributor, I grew a great deal over this release due to the cross-team negotiating and coordination required for success, along with leading the team more broadly while my PM lead peer took

leaves before departing Microsoft (requiring me to step up and run adds /cuts for my team along with filling in for PM leadership).

Delivered: the features that my team implemented this release include: Document Information Panel integration of hosted InfoPath into Word, Excel, and PowerPoint; integration of InfoPath email forms into Outlook; WSS rewrite of InfoPath form libraries to be heterogeneous site content-type document libraries; integration of InfoPath into Office workflow feature; integration of InfoPath designer into Visual Studio 2005 as a hosted designer; integration of Visual Studio Tools for Applications into InfoPath for C# and VB.Net business logic authoring; componentization of InfoPath editor as a 3rd party hostable control; and load balanced forms server features like rules, calculations, conditional formatting, and rich-text editing.

Customer focus: at the beginning of the product cycle, I made a business case for the InfoPath team blog and ushered it through its early months, providing quick, quality content for our user base to solve their issues and understand the immense benefits that InfoPath 2003 SP1 provided. I also coordinated with Robert Scoble to get a video discussing InfoPath 2007 client and server features onto Channel9.

Microsoft Office InfoPath 2003 SP1 – Development Lead / Lead-of-leads
2003 – 2004

I led a team of developers to add user-focused *Information-Worker* specific features to InfoPath 2003 SP1. We quickly and efficiently drove end-to-end feature ownership to add high-impact features such as:

Calculations: Declarative, easy-to-build- XPath expressions for performing calculations and storing results back in the XML. Users no longer have to use events and script.

Rules: Simple conditions to either change the XML or perform end-user actions.

TabletPC: Users are now able to just ink directly on-top of the form and have their hand-writing converted. We also added ad-hoc insertable ink sections.

Roles: Upon load, the appropriate for role the form user can be ascertained and used within the rules and business logic to customize the form filling experience for the user based on their role.

Signed full-trust forms: To simplify the distribution of full-trust forms (forms that are capable of doing actions similar to an application), my team added the ability to sign a form and allow the user to trust such a form on their machine.

Although labeled "SP1" this release added significant features to the product. I assumed the role as the dev team's bug / triage guy and represented the team at Office's SP1 triage meetings, ensuring our changes were not disruptive to the patch.

Microsoft Office InfoPath 2003 – Development Lead
2001 – 2003

I was a front-line manager for the V1 InfoPath 2003 release, leading a team of developers to implement the design-time and run-time aspects of the controls that comprise an InfoPath form template.

In addition to managing (including leading the transition of WDEs to SDEs), I was deeply involved in the design and implementation of the control architecture along with other key aspects of the InfoPath designer. I implemented a variety of features in addition to running the normal scheduling and team management duties.

I also kept in close contact with the "out of the box" solutions team, minding their dependence on my team's controls.

Microsoft NetDocs – Development Lead + Individual Contributor (Developer)
1997 – 2001

NetDocs was a grand vision of a software-as-a-service platform and a subscription-based application suite. It used XML technology very heavily, including XSLT and schema (XDR), along with the XMLHttpRequest object for replicated WebDav data collection (it was Ajax on steroids).

I started as an individual contributor, responsible for NetDocs specific C++ / COM based XSL engine (back when XSL was new). I eventually led the transition to the MSXML engine's XSLT implementation.

As a front-line dev lead, I was responsible for the team that delivered collection view technology and that delivered end-user features using the collection views, including email, calendaring, and contacts. Our collection views were completely asynchronous (like all of NetDocs) and provided virtualized views with virtualized selection.

The structured XML editing abilities of NetDocs were salvaged to become InfoPath.

Intel Supercomputer Systems Division – Individual Contributor (Developer)
1992 – 1997

I was the user interface expert for the graphical development tools delivered for Intel's TFLOP machine and the Paragon series of Supercomputers.

My main responsibility was the design and development of XIPD, the graphical front-end to the command line debugger. The final multi-platform incarnation, written in C++ using the Motif widget set, divided the front-end and back-end via hand-crafted RPC calls.

Seeing an issue with system crash analysis, I delivered a Perl script that issued kernel-debugger commands on a crashed super-computer and traversed through the nodes, based on pattern matching rules, to track down what the issue might be. This helped with field evaluation of bad OS software (or bad hardware).

I was also the HTML / Web evangelist in 1993, after playing with Viola then Mosaic. I jump-started my group's usage of HTML, contributing to a FrameMaker to HTML converter, and coordinated cross-company with the team getting Intel's first web presence delivered.

EPOS Corporation – Development Manager, Individual Contributor (Developer)
1987 – 1989

In this aggressive startup-up company, I worked with the QNX real-time operating system as a platform for creating plastics-injection molding monitoring software. Early on, I convinced the team to develop our user-interface library using a function-pointer-based rolled-by hand version of object-oriented programming (allowing us to implement a VT-based renderer within minutes).

Growth: dealing with a message-passing OS, 64K code segments, customer requirements for quality-control addition, dealing effectively with customers face-to-face, field install of software and hardware, and the day-to-day drama of working for a company constantly flirting with financial ruin. I passed the Engineer In Training (EIT) exam and I was working towards my Professional Engineer certification.

Education

Auburn University, Masters Computer Science
June 1992

Emphasis of my Masters study was user interface design (including understanding basics of cognition and physiology) and computer language theory.

Thesis: *HC: A Generator for Platform Independent Graphical User Interfaces*. I designed the *hc* language to describe an application's platform independent user interface. Via lex and yacc, the *hc* code was parsed and

turned into C++ and resource files that utilized multiple-inheritance to bind platform specific behavior (either Macintosh or X11) to the generic user interface library that I wrote.

Auburn University, Bachelors of Computer Engineering
June 1987

Computer Science with a wide variety of engineering courses instead of a minor. With honors.